

AUTOMATED ORBIT DETERMINATION SYSTEM (AODS)
REQUIREMENTS DEFINITION AND ANALYSIS

Prepared for
GODDARD SPACE FLIGHT CENTER

By
COMPUTER SCIENCES CORPORATION

Under
Contract NAS 5-24300
Task Assignment 989

Prepared by:

Sharon R. Waligora 9/30/80
S. R. Waligora Date

Approved by:

C. E. Goorevich 9/30/80
C. E. Goorevich Date
Technical Reviewer

Arthur L. Green 9/30/80
A. L. Green Date
Section Manager

R. D. Headrick 9/30/80
R. D. Headrick Date
Department Manager

ABSTRACT

This document provides the requirements definition and analysis for the prototype version of the Automated Orbit Determination System (AODS) currently being developed at Goddard Space Flight Center's System Technology Laboratory (Code 580). Specified herein are the AODS requirements at all levels, the functional model as determined through the structured analysis performed during requirements definition, and the results of the requirements analysis. Also specified are the implementation strategy for AODS and the AODS-required external support software system (ADEPT), input and output message formats, and procedures for modifying the requirements presented in this document.

TABLE OF CONTENTS

<u>Section 1 - Introduction</u>	1-1
<u>Section 2 - Requirements Definition</u>	2-1
2.1 System Requirements Definition	2-2
2.1.1 Top-Level Requirements	2-2
2.1.2 Input Requirements	2-5
2.1.3 Output Requirements	2-7
2.2 System Performance Requirements and Limitations Definition	2-8
2.2.1 System Performance Requirements	2-8
2.2.2 Hardware and Software Requirements and Limitations	2-10
2.3 Functional Requirements Definition	2-11
2.3.1 Functional Requirements	2-11
2.3.2 AODS Functional Model	2-19
2.3.3 Requirements Maps	2-36
2.4 Functional Specifications	2-37
2.4.1 System Control Functional Specifications	2-37
2.4.2 Input Processing Functional Specifika- tions	2-41
2.4.3 Data Preprocessing Functional Specifi- cations	2-42
2.4.4 Data Management Functional Specifika- tions	2-44
2.4.5 Estimation Functional Specifications	2-44
2.4.6 One-Way Doppler Functional Specifika- tions	2-47
2.4.7 Output Processing Functional Specifi- cations	2-48
<u>Section 3 - Requirements Analysis</u>	3-1
3.1 Operational Scenario	3-1
3.2 Timing and Computer Memory Studies	3-5
3.2.1 Timing Study	3-5
3.2.2 Computer Memory Requirements	3-13

TABLE OF CONTENTS (Cont'd)

Section 3 (Cont'd)

3.3	Throughput Analysis	3-16
3.3.1	Worst Case Throughput Analysis.	3-18
3.3.2	Typical Case Throughput Analysis.	3-21
3.3.3	Conclusions and Recommendations	3-23
3.4	Error Contingency Plan.	3-25
3.4.1	Computational Errors.	3-25
3.4.2	Algorithmic Errors.	3-25
3.4.3	Input Data Errors	3-26
3.4.4	Operational Errors.	3-26

Section 4 - AODS Development Strategy

4.1	Current Activities.	4-3
4.1.1	AODS Implementation	4-4
4.1.2	ADEPT Implementation.	4-12
4.2	Future Builds	4-18

Appendix A - Structured Analysis

Appendix B - Data Dictionary

Appendix C - Input and Output Message Formats

Appendix D - Procedures for Modifying AODS Requirements

References

LIST OF ILLUSTRATIONS

Figure

2-1	AODS Context Diagram.	2-3
2-2	Top Level of AODS	2-20
2-3	Process 1: System Control.	2-21
2-4	Process 2: Input Processing.	2-22
2-5	Process 3: Data Preprocessing.	2-23
2-6	Process 4: Data Management	2-24
2-7	Process 4.1: TDRS Orbit File Management.	2-25
2-8	Process 4.2: Observation File Management	2-26
2-9	Process 5: Estimation.	2-27
2-10	Process 5.1: Estimation Initialization	2-28
2-11	Process 5.2: Accumulation of Statistics From Observations.	2-29
2-12	Process 5.6: Report Generation	2-30
2-13	Process 6: One-Way Doppler Prediction.	2-31
2-14	Process 7: Output Processing	2-32
4-1	GSFC-Specified Schedule for AODS Development	4-2
4-2	AOPS/ADEPT First-Build Internal Communi- cations Design.	4-6
4-3	Executive Control	4-9
4-4	Output Preparation.	4-11
4-5	ADEPT Overview.	4-14

LIST OF TABLES

Table

2-1	Functional Requirement/Process Map.	2-38
2-2	Process/Functional Requirement Map.	2-39
2-3	Service Control Commands and Results.	2-40
3-1	Comparison of PDP-11/70 and LSI-11/23 Instruction Times	3-7
3-2	PDP-11/70 Timing Test Results	3-9
3-3	Execution Times of Existing Software Components.	3-10
3-4	Input Message Transmission Times.	3-12
3-5	Output Message Transmission Times	3-12
3-6	AODS Data Storage Requirements.	3-14
3-7	Estimated Size of AODS Software Components.	3-15

SECTION 1 - INTRODUCTION

This document provides the requirements definition and analysis for the prototype version of the Automated Orbit Determination System (AODS) currently being developed at Goddard Space Flight Center's (GSFC's) System Technology Laboratory (STL), Code 580. This prototype system will demonstrate, in a laboratory environment, the feasibility of using microprocessors to perform onboard orbit determination in an automated manner with limited ground support. It is anticipated that this prototype system will be used as the basis for an experimental flight system to be flown on a mid-1980 mission. The orbit determined on board could be used to provide both position and velocity for scientific experiment data annotation and predicted one-way Doppler data for receiver acquisition. This document also includes a discussion of the external support software required by AODS. This software system, named ADEPT for AODS Environment Simulator for Prototype Testing, provides the external information required for AODS operation and monitors AODS performance. The system requirements of the actual flight experiment are not addressed in this document.

AODS will be developed on STL's PDP-11/70 computer under the RSX-11M operating system. The target computer will be the LSI-11/23 microprocessor under the RSX-11S operating system. Since the LSI-11/23 supports FORTRAN IV Plus and double-precision arithmetic, it is well suited for complex computational problems such as orbit determination. ADEPT will be fully implemented on the PDP-11/70.

Section 2 of this document specifies the AODS requirements at all levels and the functional model as determined through the structured analysis (Reference 1) performed during requirements definition. Section 3 presents the results of the requirements analysis, including timing and sizing

studies, operational scenario, throughput analysis, and error contingency plan. Section 4 specifies the implementation strategy for AODS and ADEPT for both current and future implementation. Appendix A states the ground rules for the data flow diagrams provided in Section 2. Appendix B contains the data dictionary that is referred to throughout the document. Appendix C specifies the input and output message formats. Appendix D specifies the procedures for modifying the requirements presented in this document.

SECTION 2 - REQUIREMENTS DEFINITION

This section specifies the AODS requirements that have been determined through the system definition phase of requirements analysis. Given a basic set of requirements and the information gathered from extensive interviews with GSFC, a formal methodology, structured analysis (Reference 1), was used to build a preliminary model of the system. Then, through a series of requirements reviews, this model was iteratively enhanced until the desired system model was obtained. The model of AODS then served as the center of discussion through which a more extensive requirements list was generated.

The AODS requirements specified in this section are presented according to level of detail, as follows:

- Section 2.1 specifies the system requirements, which are the tasks the system must perform (on the highest level) to satisfy the needs and objectives of the end user.
- Section 2.2 specifies the system performance requirements and limitations, which consist of the schedules on which specific requirements must be satisfied and any limitations that will affect the performance of the system.
- Section 2.3 specifies the functional requirements, which are the functions the system must perform to satisfy the system requirements. These are the most detailed requirements given. In addition, the section provides a functional decomposition of the system, corresponding data flow diagrams, and requirements maps to demonstrate the completeness of the AODS functional model and provide a cross-reference of the functional requirements.

- Section 2.4 presents the functional specifications, which are the computational models and procedures that have been specified to date.

2.1 SYSTEM REQUIREMENTS DEFINITION

AODS will be an onboard orbit determination system requiring periodic ground support. The objective of AODS is to provide the outside world with orbit information (i.e., position and velocity) on a near-real-time basis for experimental data annotation.

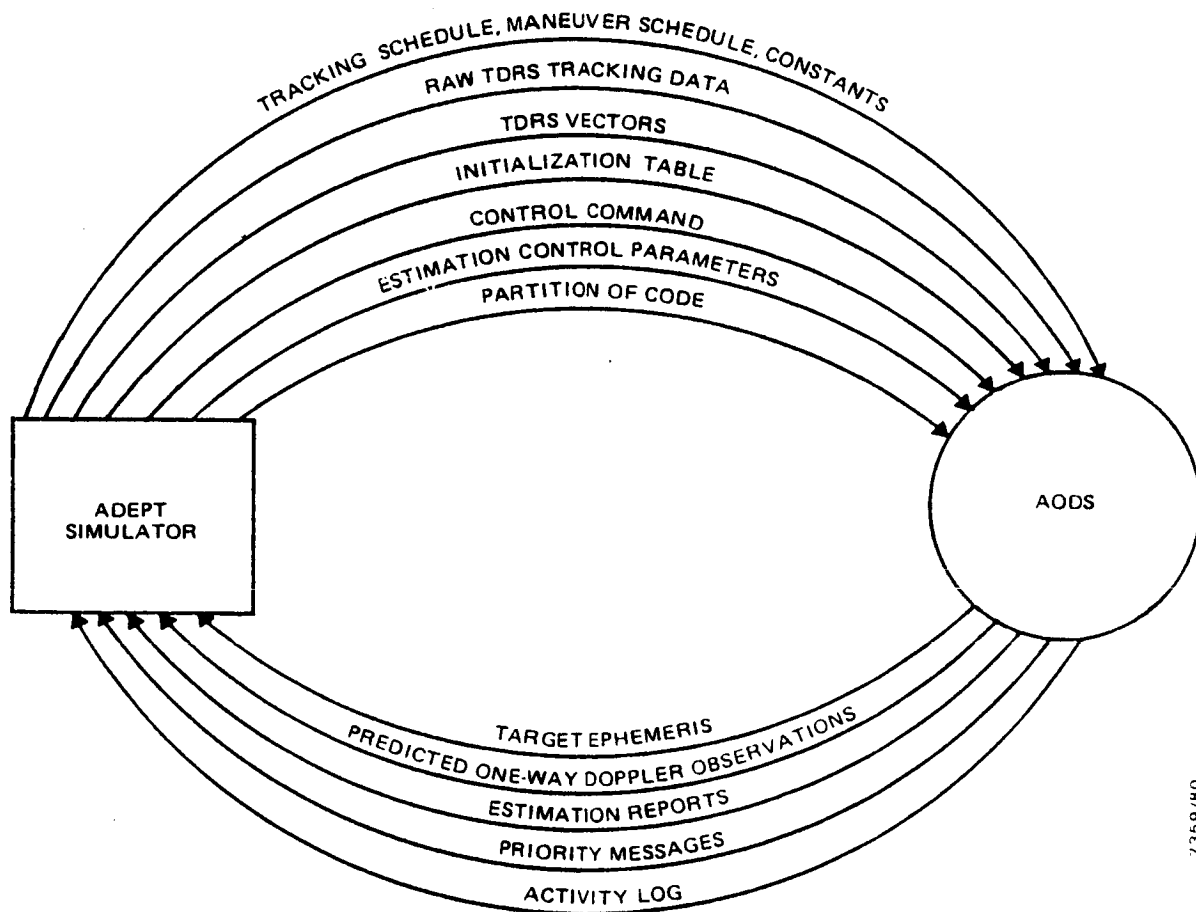
Since the prototype AODS will be built and demonstrated in a laboratory environment, the external world, including the ground support system, will be simulated by ADEPT. Figure 2-1, the AODS context diagram, shows the relationship of the prototype AODS to its external environment.

This section specifies the system requirements, i.e., the tasks that the prototype AODS must perform to satisfy the needs and objectives of the end user. These requirements include the top-level AODS requirements, presented in Section 2.1.1, and the input and output requirements, presented in Sections 2.1.2 and 2.1.3, respectively.

2.1.1 TOP-LEVEL REQUIREMENTS

The top-level requirements of AODS are as follows:

- AODS will provide position and velocity on a near-real-time basis for experimental data annotation and direct downlink.
- AODS will predict one-way Doppler observations on a scheduled basis and output them for receiver acquisition.
- AODS will generate and output a state vector predict table containing vectors at a specified frequency over a specified time interval.



7359/80

Figure 2-1. AODS Context Diagram

- AODS will maintain and output an activity log on a regular basis and when specifically requested through a control command.
- AODS will perform any preprocessing required to process the input raw tracking data.
- AODS will be capable of recovering from both user spacecraft and Tracking Data and Relay Satellite (TDRS) maneuvers.
- AODS will perform orbit determination using a batch least-squares method of estimation, differentially correcting the orbit of the target (user spacecraft). AODS will estimate the following state parameters:
 - Six parameters of the orbital state (target) (position and velocity)
 - Atmospheric drag coefficient, C_D
 - Coefficients of the frequency model for one-way TDRS System (TDRSS) data

AODS will process the following types of observation data:

- One-way TDRSS range and Doppler
- Two-way TDRSS range and Doppler
- Hybrid TDRSS range and Doppler
- One-way Standard Ranging Equipment (SRE) range and range-rate
- Two-way SRE range and range-rate

2.1.2 INPUT REQUIREMENTS

The AODS input requirements are as follows:

- AODS will accept input messages containing data, executable code, and control commands.
- AODS input data will consist of the following:
 - Raw observation messages--These messages are sets of raw observations in the universal tracking data format. One observation set consists of a maximum of 100 observations (pairs of range and Doppler measurements) that were taken during the same spacecraft pass or contact.
 - New TDRS vectors--These data include one state vector (position and velocity) for each active TDRS, up to three TDRSs. New TDRS vectors will be uplinked at least once per day.
 - Maneuver schedule--This schedule specifies the predicted states and times of user spacecraft and/or TDRS maneuvers. It covers up to 16 maneuvers and will be uplinked as necessary. The entire maneuver schedule will be uplinked at the same time.
 - Tracking schedule--This schedule is the tracking schedule for the prediction of one-way Doppler observations. It covers 16 tracking intervals and will be uplinked as necessary. The entire tracking schedule will be uplinked at the same time.
 - Initialization table--This table specifies the initial conditions for the estimator, including the a priori state vector, which will be propagated for output until a solution is

reached. This table will be uplinked at the start of AODS execution and then later at the user's discretion.

- Constants--These constants, which will be used throughout the AODS processes, may have to be changed during long-term operations. They are categorized as follows: integration, conversion, and physical constants; station positions (20 stations) and observation modeling constants; geopotential model constants; atmospheric drag model constants; and timing coefficients.
- Estimation control parameters--This set of parameters (e.g., maximum iterations, observation weights, convergence criteria) provides control in estimating the spacecraft state. It will be uplinked at the first estimation process and then later at the user's discretion.
- AODS will recognize the following control commands:
 - REBOOT--Reboot AODS.
 - ABORT--Abort AODS processing; output activity log; do not destroy code.
 - STOP--Terminate AODS processing in a normal manner; do not accept more data.
 - START--Start AODS processing; accept all data. (This is a reply to commands STOP and ABORT.)
 - SUSPEND--Suspend orbit determination process; continue accepting data.

- CONTINUE--Resume orbit determination computations. (This is a reply to command SUSPEND.)
- STATUS REQUEST--Output activity log.
- SET CLOCK--Set system clock to new time.
- AODS will accept and load a partition of executable code.

2.1.3 OUTPUT REQUIREMENTS

The AODS output requirements are as follows:

- AODS will periodically output an activity log containing a history of all activities that have been performed by AODS.
- AODS will output priority messages to request special ground support such as error handling, fast-timing, and so forth.
- AODS will output tables of predicted state vectors to the main onboard computer for experimental data annotation and direct downlink.
- AODS will output predicted one-way Doppler observations on a scheduled basis to the main onboard computer for receiver acquisition.
- AODS will output the following reports from the estimator:
 - DC Residuals Report--This report contains information about each individual observation (e.g., tracking configuration, observation residual, editing).
 - DC Summary and Statistics Report--This report contains differential correction (DC) summary information (e.g., state update, new state,

standard deviations of state parameters) and DC statistics (e.g., current root-mean-square (rms), previous rms, batch editing statistics).

2.2 SYSTEM PERFORMANCE REQUIREMENTS AND LIMITATIONS DEFINITION

This section specifies those requirements that deal with system performance and the limitations associated with it. Section 2.2.1 presents the system performance requirements that define the schedules on which specific requirements must be satisfied. Section 2.2.2 presents the hardware and software requirements and the limitations that will affect AODS performance.

2.2.1 SYSTEM PERFORMANCE REQUIREMENTS

The system performance requirements for AODS are as follows:

- AODS will capture all incoming messages upon demand.
- AODS will service each control command immediately after reception.
- AODS will maintain an activity log and output (downlink) it on a scheduled basis or when requested by a control command.
- AODS will output a table of predicted user spacecraft state vectors over a specified time interval at a specified frequency. For example, if the time interval is 1/2 hour and the frequency is 1 minute, the state vector predict tables will be generated as follows:
 - Each time a new solution is reached or a new a priori state vector (initialization table) is received, a table containing state vectors at 1-minute intervals starting at the current time (t_n) and ending 1 hour later ($t_n + 1$) will be generated and output.

- Then, 1/2 hour later ($t_n + 1/2$), the next table will be generated and output. This table will contain state vectors at 1-minute intervals over the next 1/2 hour. The start time of this table will be the end time of the previous table ($t_n + 1$), and the end time will be 1/2 hour after that ($t_n + 1 - 1/2$).
- The second step will be repeated until a new solution is reached or a new a priori state vector is received, which causes the process to begin again with the first step.
- AODS will output one-way Doppler observations no later than 1 minute before the start time of the current tracking interval. The actual amount of lead time will be set by the ground control.
- AODS will complete data preprocessing and estimation on each batch of data by the time the next pass of raw observations is received. Since observation data will be transmitted (uplinked) every revolution under normal circumstances, this processing time will be limited to the length of one revolution of the user spacecraft (nominally, 90 minutes).
- AODS will be capable of performing batch estimation over a user-specified time span, which will never be larger than 50 hours. In addition, AODS must be capable of handling a maximum of 500 observations in each batch of data.
- AODS will be capable of generating two types of reports during differential correction:
 - The DC Residuals Report, if generated, will be generated either (1) after the first and last

iterations on each batch of data or (2) after the last iteration on each batch of data.

- The DC Summary and Statistics Report, if generated, will be generated either (1) after the last iteration on each batch of data or (2) after every DC iteration.

2.2.2 HARDWARE AND SOFTWARE REQUIREMENTS AND LIMITATIONS

The AODS hardware and software requirements and the limitations associated with them are as follows:

- The development computer will be STL's PDP-11/70 under the RSX-11M operating system.
- The target computer will be an LSI-11/23 under the RSX-11S operating system. It will have 256K bytes of random-access memory (RAM). No peripherals will be available.
- All necessary system software (i.e., the device handlers) in both the development and target computers will be available.
- Since there will be no peripherals in the target system, all data must be managed in RAM. In addition, overlaying of tasks is impossible.
- Since the prototype AODS could lead to onboard and ground-based experimental systems, an effort should be made during design to separate the general functions (i.e., those which could be used in both systems) from the specific functions (i.e., those which apply only to the onboard application). This may cause these systems to execute more slowly than if they were coded for one specific application.

2.3 FUNCTIONAL REQUIREMENTS DEFINITION

This section specifies the AODS functional requirements, i.e., the functions that the system must perform to satisfy the system requirements and the performance requirements. These functional requirements are the most detailed requirements presented in this document. For further clarification, this section includes the functional model of AODS, i.e., the functional decomposition of the functional requirements and the corresponding data flow diagrams. In addition, the section contains two requirements maps to demonstrate that the functional model of AODS satisfies all AODS functional requirements and, conversely, that the functional requirements are complete.

2.3.1 FUNCTIONAL REQUIREMENTS

The AODS functional requirements specified in this section are presented according to functional areas, as follows:

- System control (Section 2.3.1.1)
- Input processing (Section 2.3.1.2)
- Data preprocessing (Section 2.3.1.3)
- Data management (Section 2.3.1.4)
- Estimation (Section 2.3.1.5)
- One-way Doppler prediction (Section 2.3.1.6)
- Output processing (Section 2.3.1.7)

These functional requirements are the most detailed requirements presented in this document. However, no attempt is made to define computational models or algorithms in this section, except where the requirements are specifically affected. The models are specified in Section 2.4.

The functional requirements specified in Sections 2.3.1.1 through 2.3.1.7 are numbered for later reference and requirements mapping. In the numbering system used, R indicates requirements and P indicates process.

2.3.1.1 System Control Functional Requirements

The functional requirements for system control are as follows:

- R1.1 AODS will maintain an activity log containing the following: system events, informative messages, error messages, directives, and control commands.
- R1.2 AODS will service each control command immediately upon reception.
- R1.3 AODS will schedule maneuver recovery according to clock time and the maneuver schedule.
- R1.4 AODS maneuver recovery will consist of the following:
 - R1.4.1 TDRS maneuver--The predicted state after the maneuver will be given to the data preprocessor to be used for future generation of the TDRS orbit file.
 - R1.4.2 User spacecraft maneuver--The TDRS orbit files and the observation file will be purged. The startup procedure will be performed; estimation will be resumed only when a complete estimation span of data has been received.
- R1.5 AODS will schedule one-way Doppler prediction a user-specified number of minutes before the start time of each tracking interval in the tracking schedule.
- R1.6 AODS will schedule the output of data and messages.
 - R1.6.1 AODS will schedule the output of severe errors from which the system cannot recover.

- R1.6.2 AODS will schedule the output of priority messages.
- R1.6.3 AODS will schedule the output of the activity log at a specified interval.
- R1.6.4 AODS will schedule the output of the activity log when specifically requested through a control command.
- R1.6.5 AODS will schedule the output of the predicted one-way Doppler observations at least 1 minute before the time tag of the first observation.
- R1.7 AODS will schedule the generation and output of the state vector predict table at the end of the specified interval after the last time of output.
- R1.8 AODS will schedule the generation and output of the state vector predict table immediately after a new solution is obtained.
- R1.9 AODS will schedule input processing when the input queue is full or when the input queue contains data and the system is otherwise idle.
- R1.10 AODS will schedule data preprocessing when a complete pass of data has been processed through input and estimation on the previous batch has been completed.
- R1.11 AODS will schedule data preprocessing when a TDRS maneuver occurs or when a new TDRS vector has been received.
- R1.12 AODS will schedule estimation when a new pass of data has been added to the observations data set.
- R1.13 AODS will notify ground control when it has an excessive amount of idle time for fast timing.

2.3.1.2 Input Processing Functional Requirements

The functional requirements for input processing are as follows:

- R2.1 AODS will capture all incoming messages upon demand.
- R2.2 AODS will accept as input messages containing data, control commands, and executable code.
- R2.3 AODS will process the following types of input data: raw observation messages, new TDRS vectors, maneuver schedule, tracking schedule, initialization table, estimation control parameters, and constants (i.e., miscellaneous constants, station constants, geopotential tables, atmospheric density tables, and timing coefficients).
- R2.4 AODS will accept the following control commands: REBOOT, ABORT, STOP, START, SUSPEND, CONTINUE, STATUS REQUEST, and SET CLOCK.
- R2.5 AODS will accept as input a partition of executable code.
- R2.6 AODS will load a partition of executable code into RAM.
- R2.7 AODS will extract the usable bytes from each raw observation message to form reduced observations and queue them by type.

2.3.1.3 Data Preprocessing Functional Requirements

The functional requirements for data preprocessing are as follows:

- R3.1 AODS will accept only those observations (raw tracking data) which have the proper tracking configuration.

- R3.2 AODS will convert the observation measurements and time tag to the correct engineering units.
- R3.3 No smoothing of the raw observation data will be performed.
- R3.4 AODS will pregenerate TDRS orbit files from the uplinked TDRS vectors (one file for each TDRS). These files will cover the same time span as the observation file; they will be used iteratively by the batch estimator.
- R3.5 AODS will update the TDRS orbit files when a new TDRS vector is received.
- R3.6 After a TDRS maneuver, AODS will use the predicted state vector as the base vector for generating the TDRS orbit file in the future.
- R3.7 After receiving an update to a TDRS maneuver, AODS will update the appropriate TDRS orbit file from the maneuver time to the current processing time by propagating the input TDRS vector.

2.3.1.4 Data Management Functional Requirements

The functional requirements for data management are as follows:

- R4.1 AODS will manage all data files in memory, since no peripherals will be provided.
- R4.2 AODS will have the capability to locate, read, and write observation records in the observation file.
- R4.3 AODS will have the capability to locate, read, write, and update the records of the TDRS orbit files.
- R4.4 AODS will have the capability to purge all data files.

2.3.1.5 Estimation Functional Requirements

The functional requirements for estimation are as follows:

- R5.1 AODS will perform differential correction on the most recent fixed-length span (specified through control parameters) of observation data.
- R5.2 The method of estimation will be batch least-squares.
- R5.3 Due to the real-time processing of AODS, the estimation time span will be slid forward to encompass each new pass of observation data. This will be referred to as a "sliding batch estimator."
- R5.4 During initialization of the estimation process (which is defined as operations included in estimation using a particular batch of data), the following will be performed:
 - R5.4.1 AODS will initialize the estimation parameters from the initialization table and/or the estimation control parameters if either was received since the beginning of the previous estimation process.
 - R5.4.2 AODS will set up the new estimation span.
- R5.5 Initialization of the estimation parameters will be performed after estimation has been suspended through a control command.
- R5.6 AODS will model the following types of observations: one-way TDRSS range and Doppler; two-way TDRSS range and Doppler; hybrid TDRSS range and Doppler; one-way SRE range and range-rate; and two-way SRE range and range-rate.
 - R5.6.1 AODS will resolve the range ambiguity during observation modeling.

- R5.7 AODS will perform n sigma editing during estimation.
- R5.8 AODS will test for convergence at the end of each iteration in the estimator.
 - R5.8.1 AODS will declare a new state solution at the point of convergence.
 - R5.8.2 AODS will stop estimation when the maximum iteration is reached, if convergence is never found.
- R5.9 AODS will be capable of generating a DC Summary and Statistics Report. This report, if generated, will be generated and output either (1) after every iteration or (2) after the last iteration on each batch.
- R5.10 AODS will be capable of generating a DC Residuals Report. This report, if generated, will be generated and output either (1) after the first and last iterations on each batch or (2) after the last iteration on each batch.

2.3.1.6 One-Way Doppler Prediction Functional Requirements

The functional requirements for one-way Doppler prediction are as follows:

- R6.1 AODS will predict (simulate) one-way Doppler observations over the time spans indicated by the uplinked tracking schedule.
- R6.2 AODS will use the TDRS, whose ID will be specified with each tracking interval, to predict the one-way Doppler observations.
- R6.3 No observation feasibility checking will be performed, since the tracking schedule will contain valid intervals for the specified TDRS.

- R6.4 The target (user spacecraft) state vector used in one-way Doppler prediction will be based on the most recent state solution. In the case in which a user spacecraft maneuver occurred or a new initialization table was received, the most recent solution will be overridden by the new a priori estimate.
- R6.5 The TDRS state vector used in one-way Doppler prediction will be based on the TDRS vector used to generate the TDRS orbit file.

2.3.1.7 Output Processing Functional Requirements

The functional requirements for output processing are as follows:

- R7.1 AODS will generate and output the state vector predict table. This table will be based on the most recent state solution. In the case in which a user spacecraft maneuver has occurred or a new initialization table has been received, the most recent solution will be overridden by the new a priori state vector.
- R7.2 AODS will output priority messages directly to the ground control.
- R7.3 AODS will output the activity log.
- R7.4 AODS will output the predicted one-way Doppler observations.
- R7.5 AODS will output the DC Residual Reports as they are generated by the estimator.
- R7.6 AODS will output the DC Summary and Statistics Reports as they are generated by the estimator.

2.3.2 AODS FUNCTIONAL MODEL

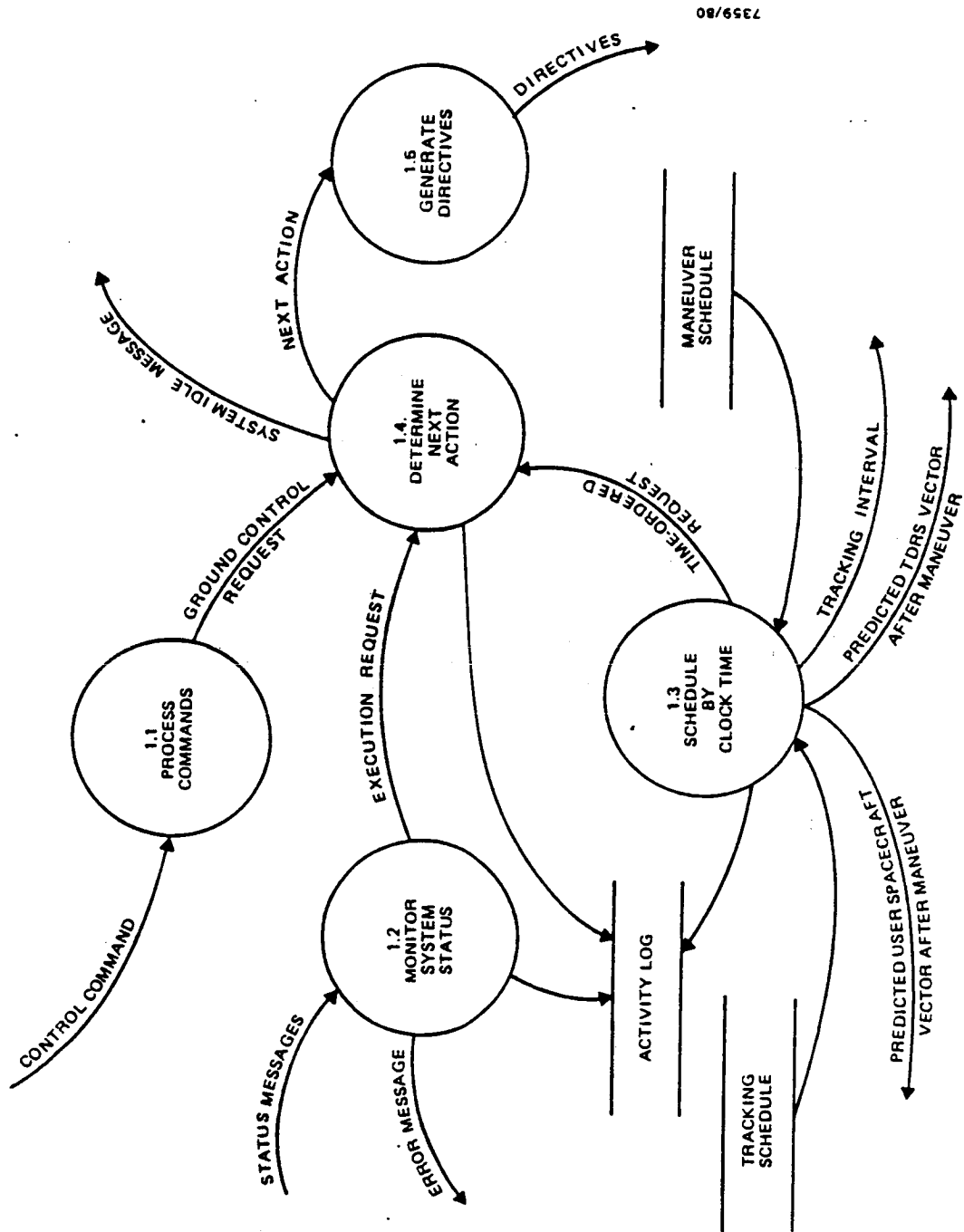
The functional model of AODS is composed of three parts: a functional decomposition of the system, the associated data flow diagrams, and a data dictionary. This section provides the AODS functional decomposition and data flow diagrams (DFDs). For the convenience of the reader, Appendix B contains the data dictionary.

The AODS functional decomposition shows the functional breakdown of each major process. The DFDs show the interfaces between processes (data flows), data stores, and data sources and sinks. Appendix A specifies the ground rules for DFDs. The data dictionary makes this model rigorous by defining each data element shown in the DFDs.

Figure 2-2 shows the top level of AODS. Figures 2-3 through 2-14 show the next two levels of detail. An asterisk by an item in any of the diagrams indicates that that item appears more than once in that DFD. Following the figures is the functional decomposition of AODS. A description of each process is provided and numbered according to the data flow diagrams. These descriptions are organized according to the seven processes specified in the top-level diagram (Figure 2-2).

The functional decomposition of the AODS is provided below. The processes are as follows:

- P1 Control AODS by monitoring system activity, scheduling by clock time, and satisfying ground commands.
- P1.1 Issue ground control requests, which are dictated by the highest priority command in the commands queue.
- P1.2 Monitor system status to determine the next action, which is dictated by normal or abnormal execution.



7359/80

Figure 2-3. Process 1: System Control

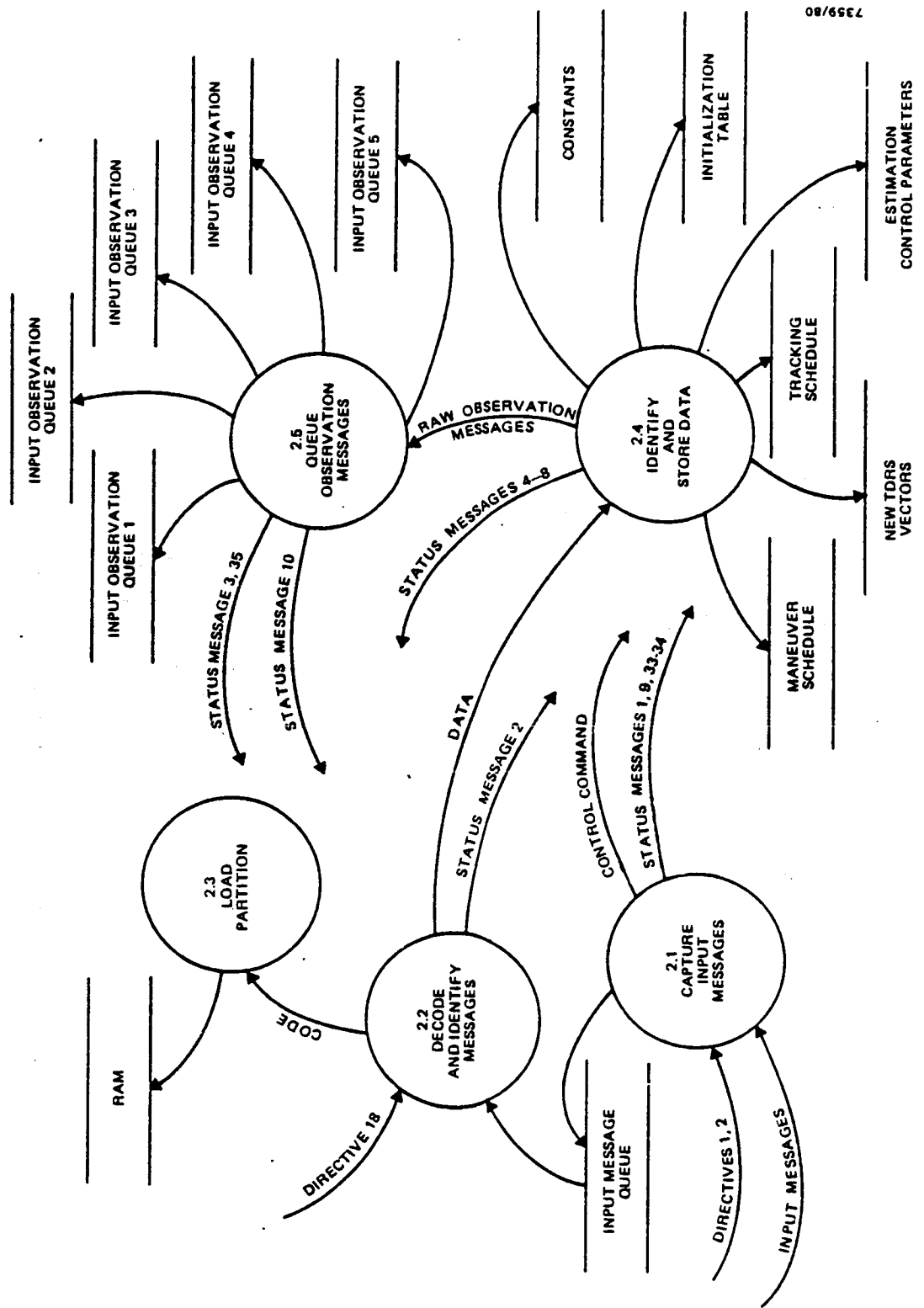


Figure 2-4. Process 2: Input Processing

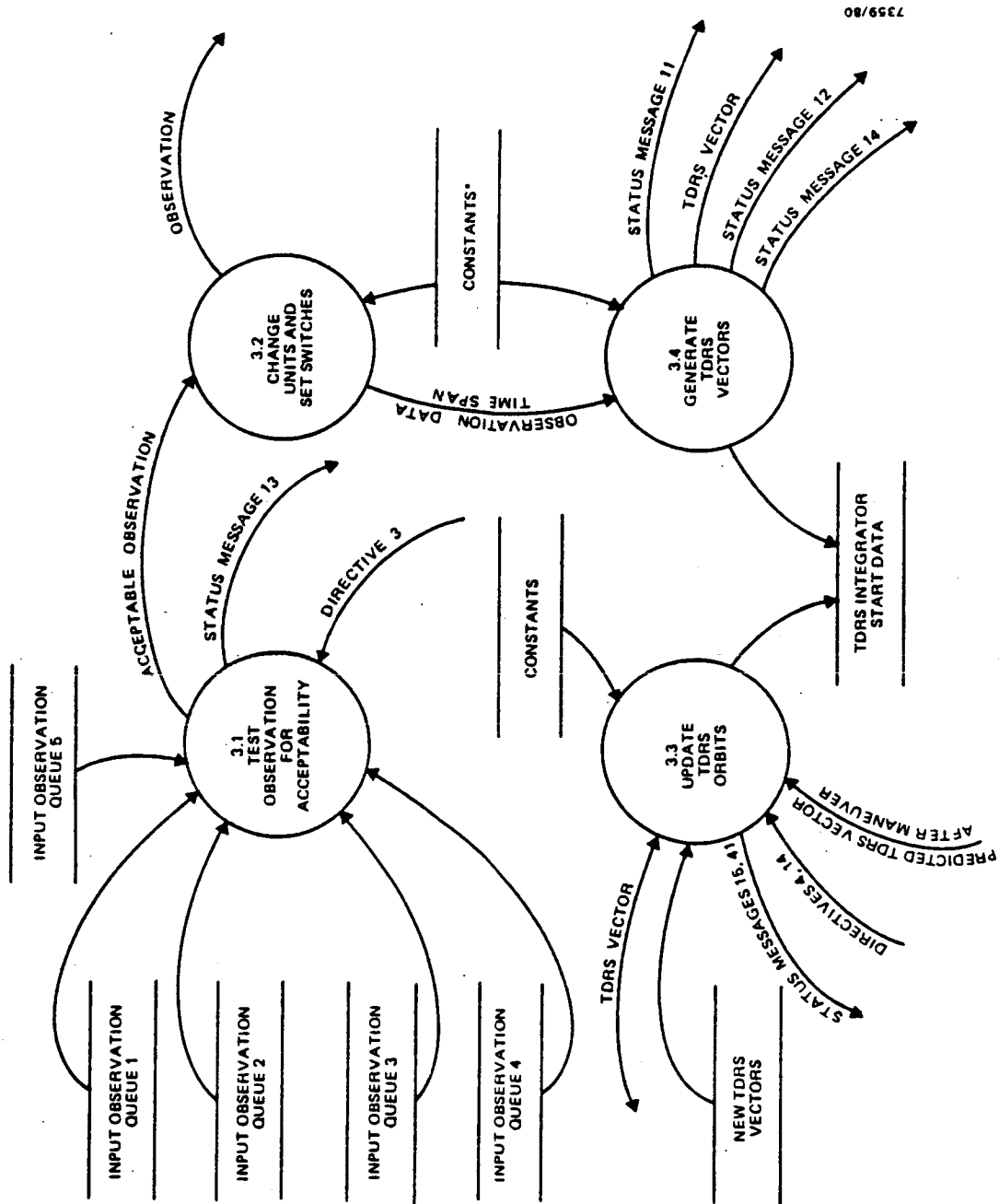


Figure 2-5. Process 3: Data Preprocessing

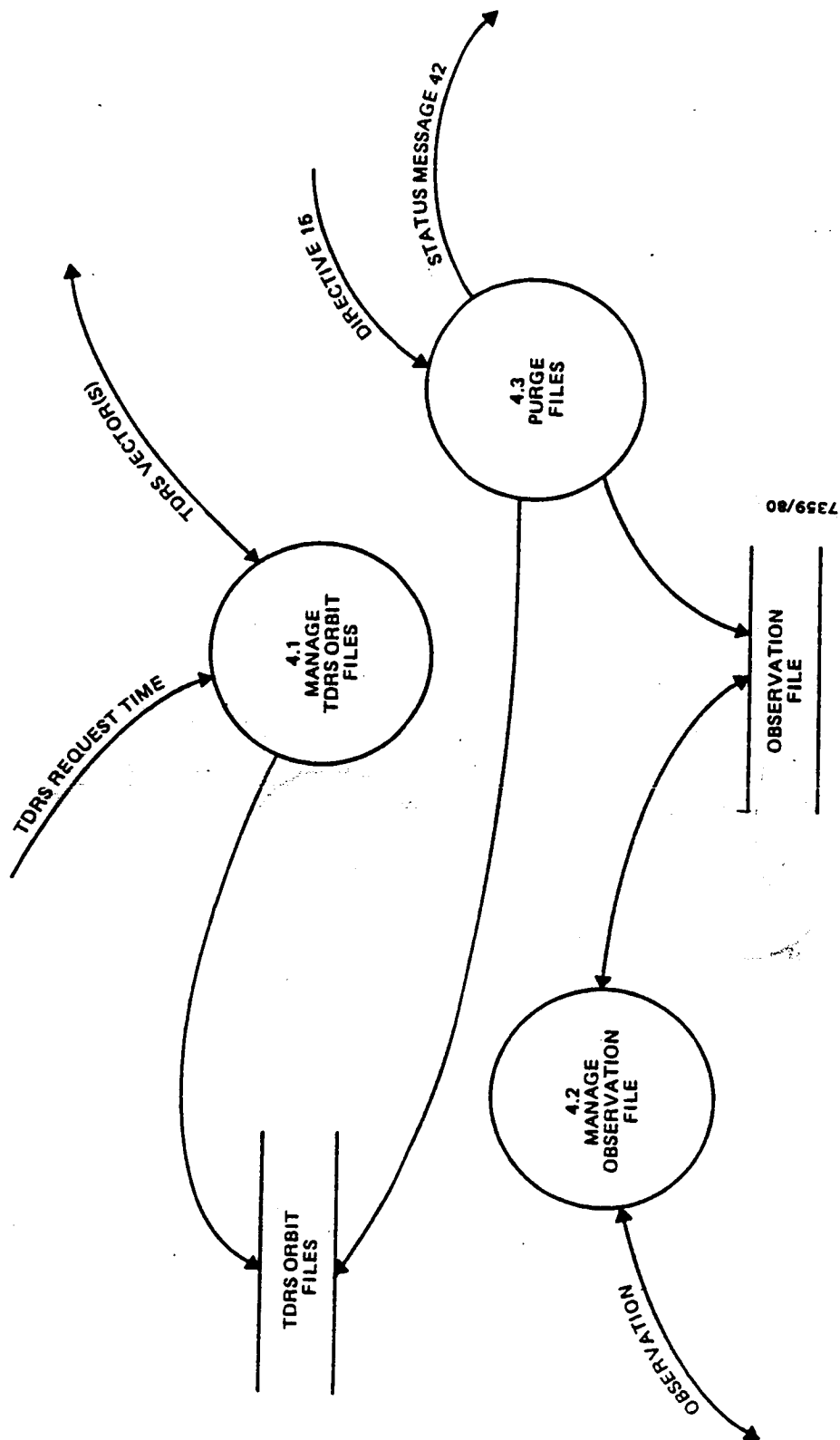


Figure 2-6. Process 4: Data Management

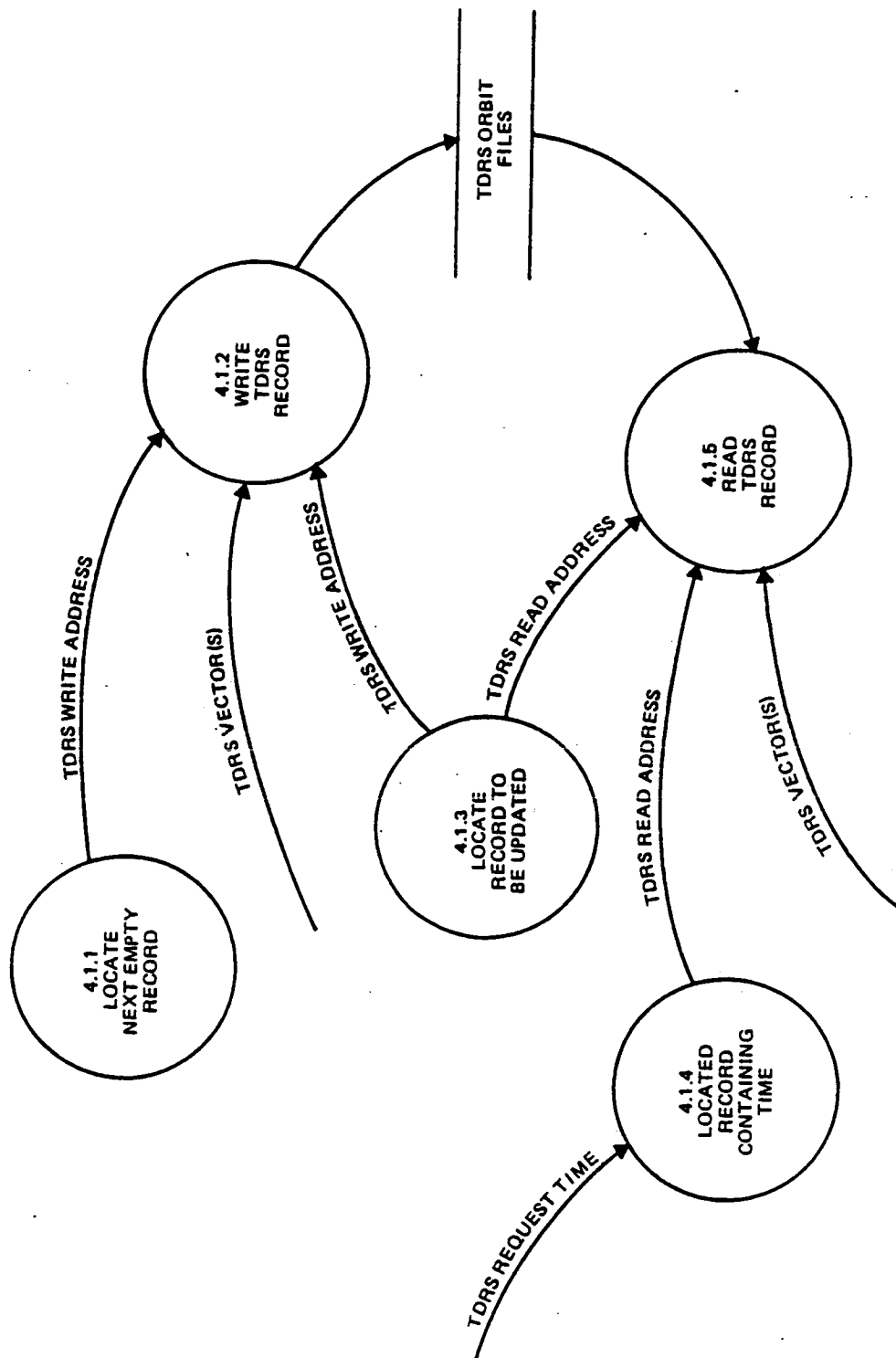


Figure 2-7. Process 4.1.: TDRS Orbit File Management

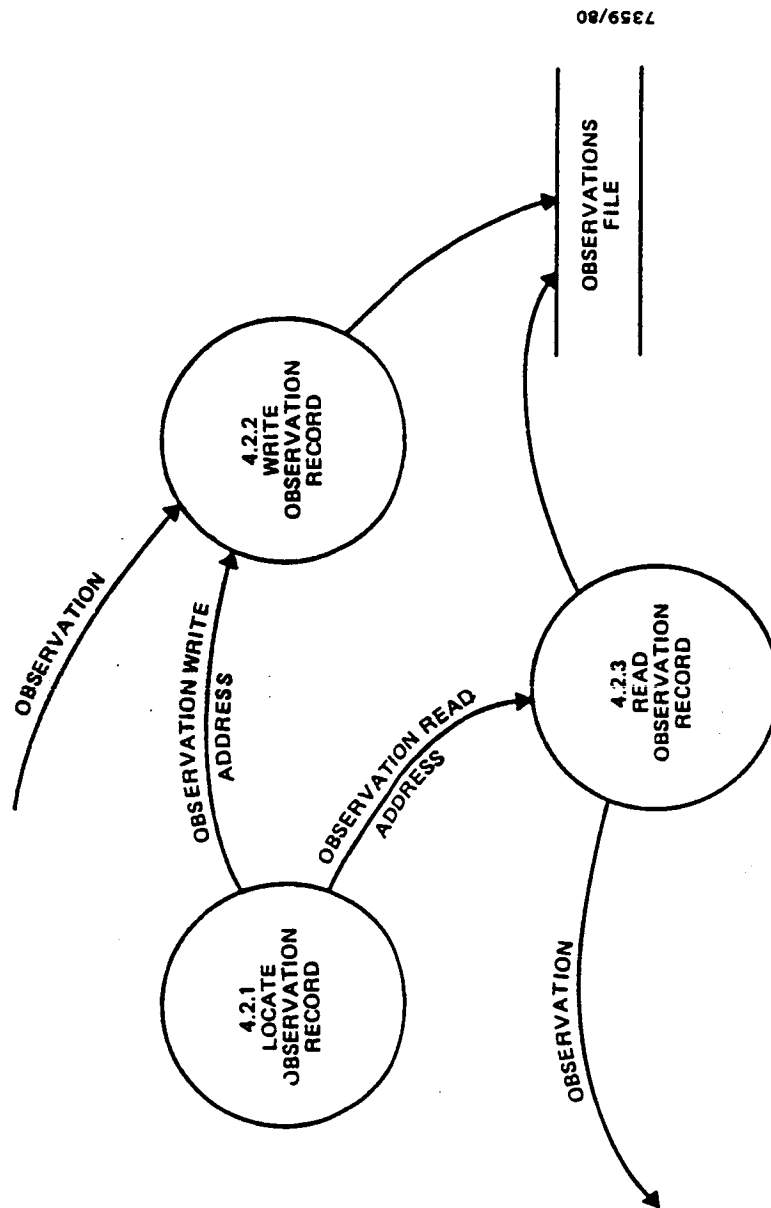


Figure 2-8. Process 4.2: Observation File Management

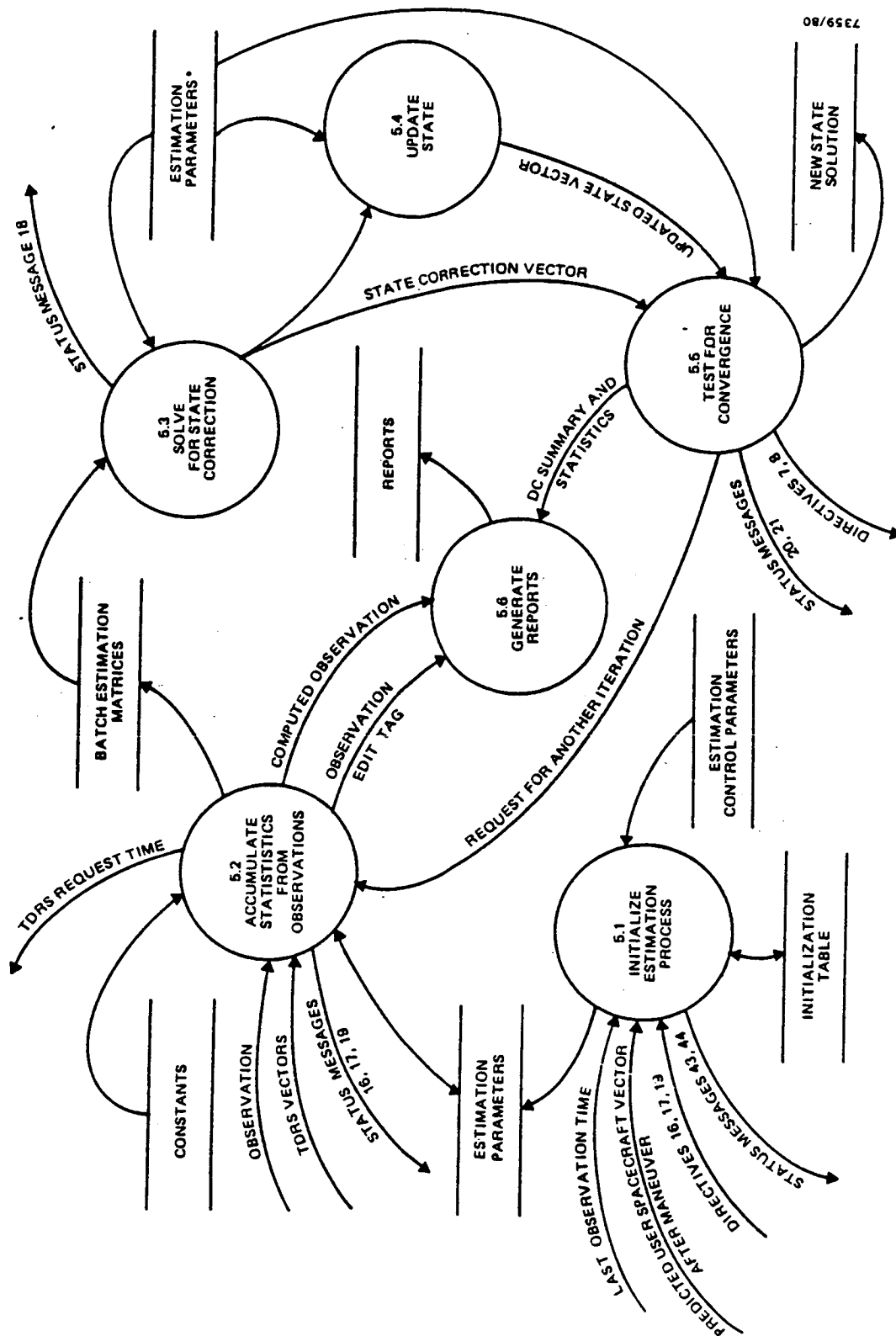


Figure 2-9. Process 5: Estimation

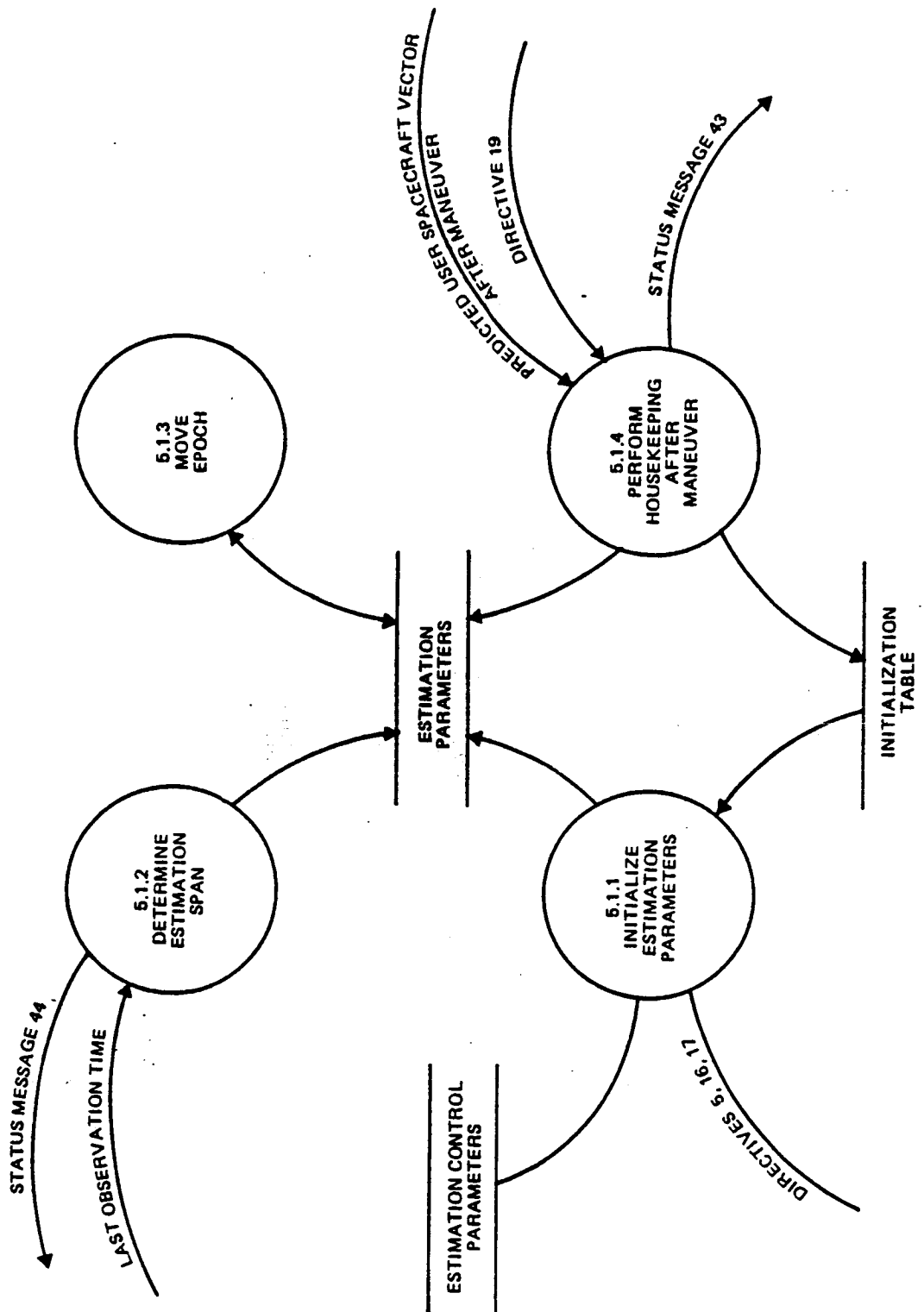


Figure 2-10. Process 5.1: Estimation Initialization

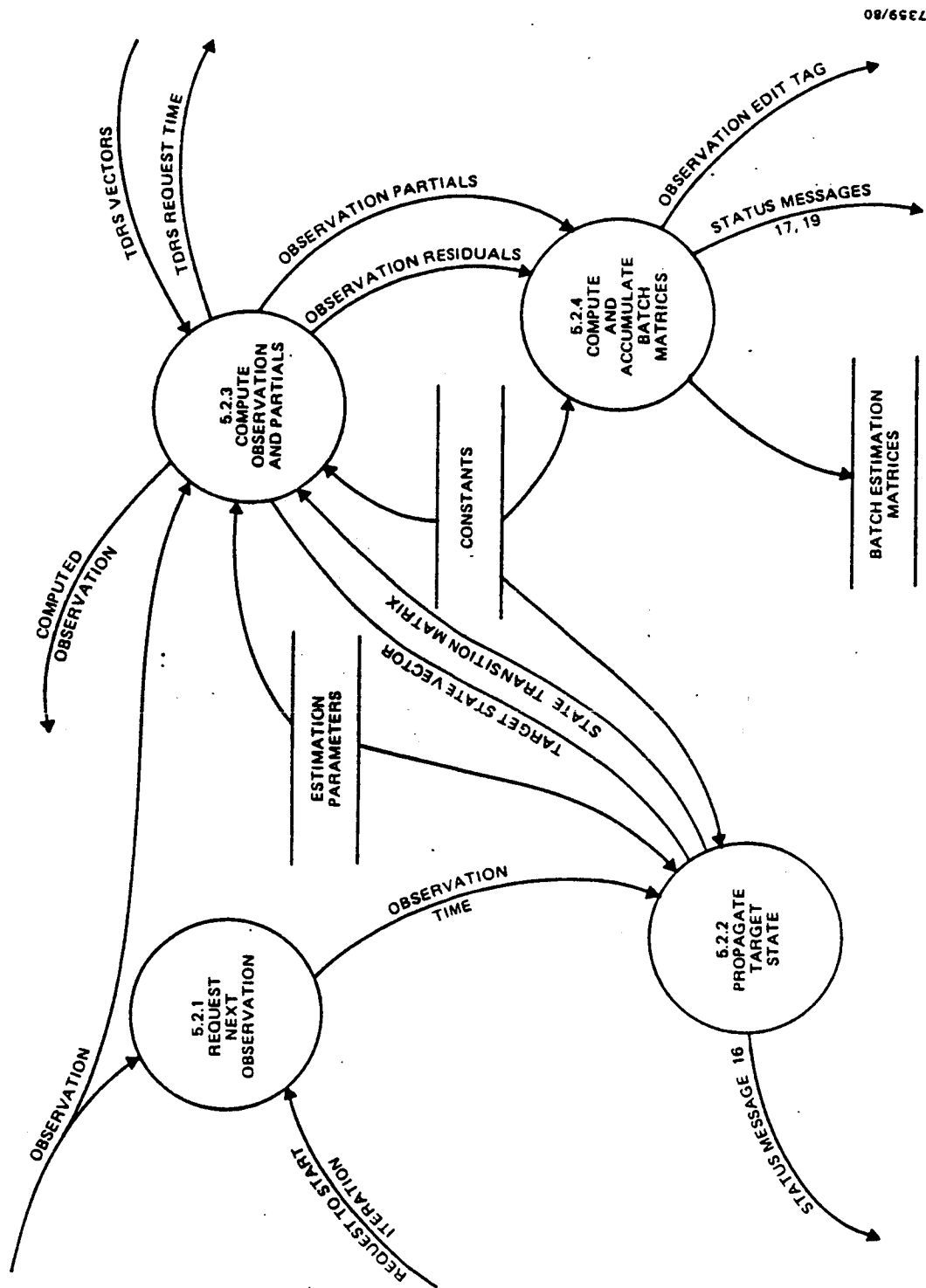


Figure 2-11. Process 5.2: Accumulation of Statistics From Observations

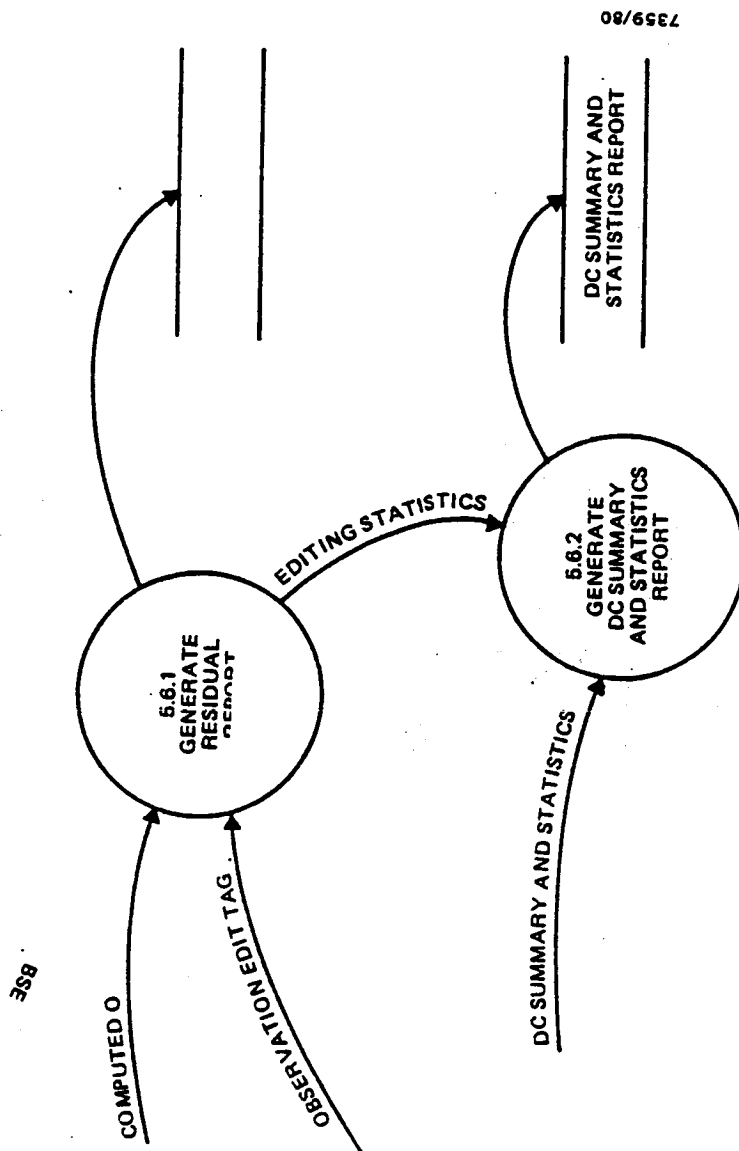
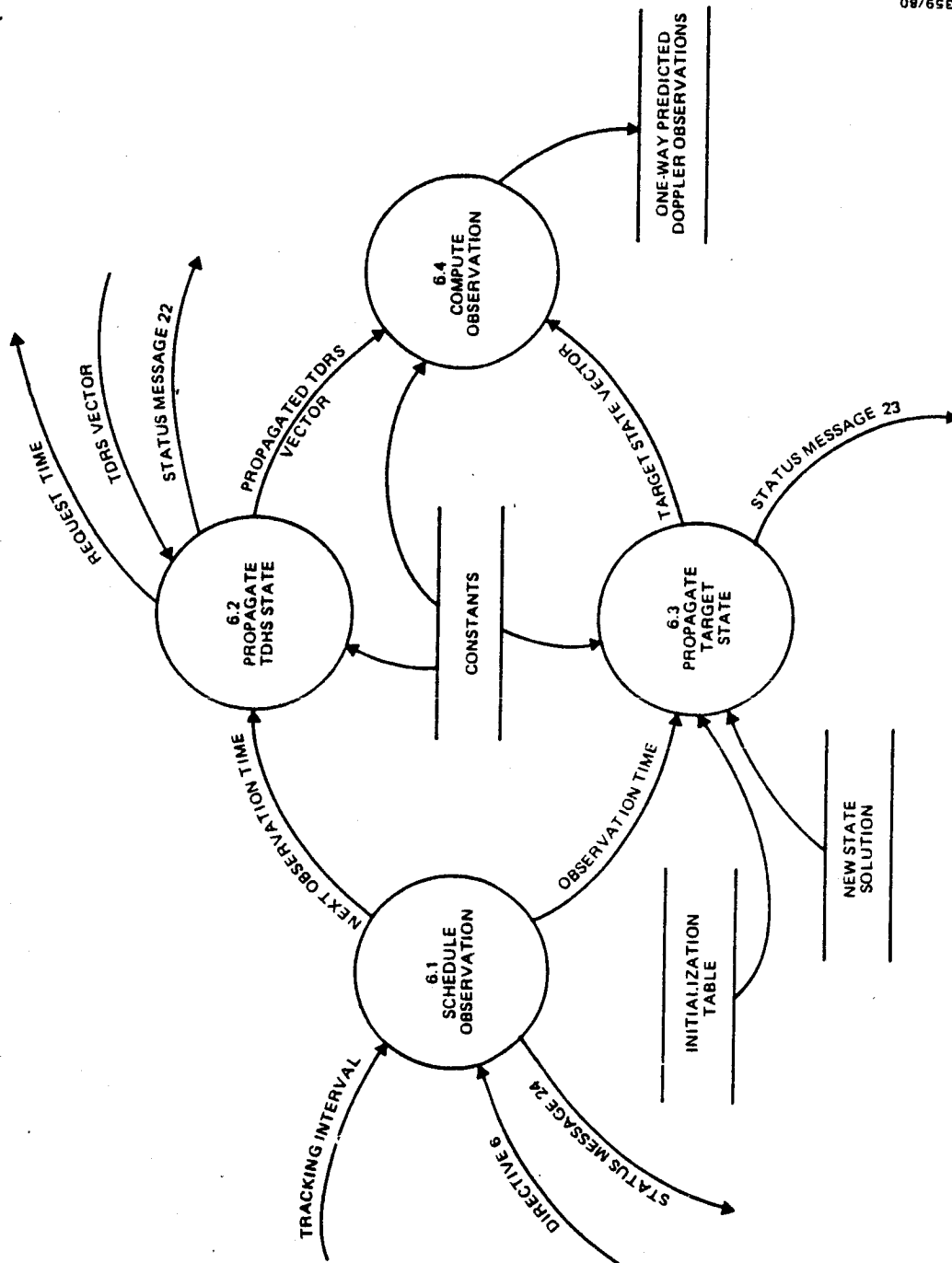


Figure 2-12. Process 5.6: Report Generation



09/65/L

Figure 2-13. Process 6: One-Way Doppler Prediction

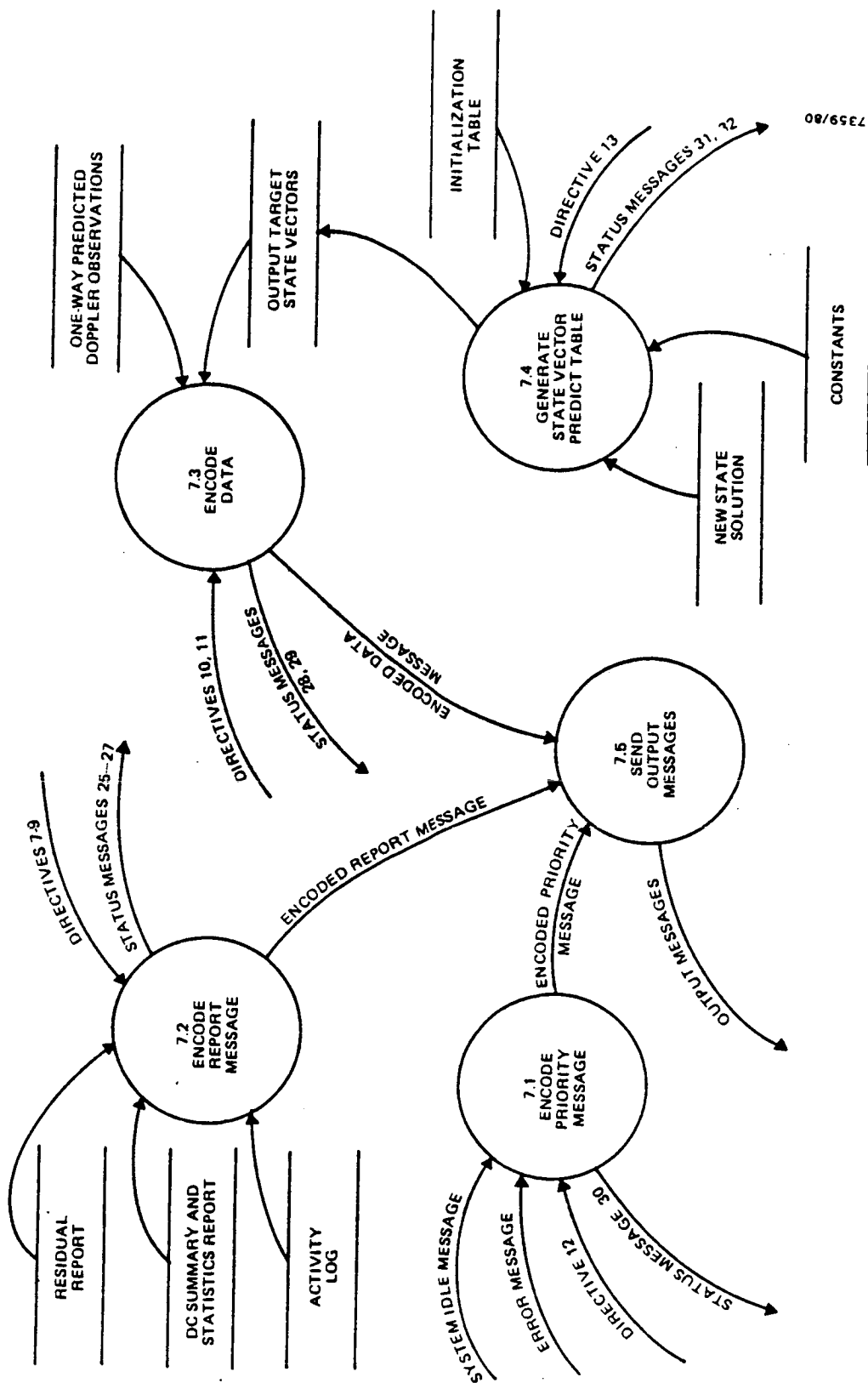


Figure 2-14. Process 7: Output Processing

- P1.3 Schedule maneuver recovery and one-way Doppler prediction according to clock time and the maneuver and tracking schedules, respectively.
- P1.4 Determine the next action based on the highest priority of the ground control, time-scheduled, and execution requests.
- P1.5 Generate the directives that will cause the action to be executed.
- P2 Receive input messages, extract and queue useful observation data, store the other data, and load program partitions into RAM.
 - P2.1 Capture and queue all input messages and send control commands to system control.
 - P2.2 Decode and identify the input messages.
 - P2.3 Load a partition of executable code into RAM.
 - P2.4 Identify and store the input data in the proper data sets.
 - P2.5 Extract the usable portions of the raw observation message and store this input observation in the proper queue based on observation type.
- P3 Prepare the observation data for later use and propagate the TDRS vectors over the observation time span. Update the TDRS orbit files when a new TDRS vector is input.
 - P3.1 Determine whether each observation has the proper tracking configuration.
 - P3.2 Change the values to engineering units and set switches that will indicate the observation's attributes.

- P3.3 Update a TDRS orbit file to reflect the new TDRS vector as follows: retrieve the next TDRS record (sequential order), propagate the new TDRS vector to the time tag, replace the TDRS state, and re-write the record.
- P3.4 Generate TDRS vectors to cover the time span of the new pass of observation data.
- P4 Manage the TDRS orbit files and the observation file.
 - P4.1 Manage the TDRS orbit files.
 - P4.1.1 Locate the next empty TDRS record.
 - P4.1.2 Write a TDRS record in memory.
 - P4.1.3 Locate the TDRS record to be updated.
 - P4.1.4 Locate the record containing the TDRS request time.
 - P4.1.5 Read a TDRS record from memory.
 - P4.2 Manage the observation file.
 - P4.2.1 Locate the next observation record in memory.
 - P4.2.2 Write an observation record in memory.
 - P4.2.3 Read an observation record from memory.
 - P4.3 Purge all files.
- P5 Estimate the state solve-for parameters based on the statistics gathered from the observed and computed observation data.
 - P5.1 Initialize the estimation process by updating the solve-for parameters from the initialization table and reinitializing switches.

- P5.1.1 Update the estimation parameters from the initialization table and estimation control parameters, if required.
- P5.1.2 Compute the start and end times of the new estimation time span.
- P5.1.3 Move epoch to the end time of the estimation time span.
- P5.1.4 Perform the necessary housekeeping after a user spacecraft maneuver.
- P5.2 Accumulate statistics from the computed and observed measurements and the current state.
 - P5.2.1 Obtain the next observation and extract the time tag.
 - P5.2.2 Propagate the target state vector to the observation time.
 - P5.2.3 Compute the observations based on the current state at the observation time and the partial derivatives of the observation with respect to the state parameters.
 - P5.2.4 Compute and accumulate the normal matrix and normal equation and perform n sigma editing.
- P5.3 Solve the normal equations to obtain the state correction vector.
- P5.4 Update the state parameters with the state correction vector.
- P5.5 Test the state and correction vectors for convergence.

- P5.6 Generate estimation reports.
 - P5.6.1 Generate a residual report line by line as observations are computed.
 - P5.6.2 Generate the DC Summary and Statistics Report at the end of each iteration.
- P6 Predict the one-way Doppler observation over a specified tracking span at a specified time interval.
 - P6.1 Schedule observations according to the tracking schedule.
 - P6.2 Propagate the TDRS state vector to the scheduled time.
 - P6.3 Propagate the target spacecraft state vector (based on the most recent solution) to the scheduled time.
 - P6.4 Compute the observation at the scheduled time based on the TDRS and target state vectors.
- P7 Propagate the target state vector for output; prepare and send the output messages for reports, data, and error messages.
 - P7.1 Encode an error message into an output message.
 - P7.2 Encode a line of a report into an output message.
 - P7.3 Encode a piece of data into an output message.
 - P7.4 Generate the state vector predict table.
 - P7.5 Send the output messages to the onboard computer (OBC).

2.3.3 REQUIREMENTS MAPS

This section presents two requirements maps that demonstrate that the functional model of AODS satisfies all AODS functional requirements, and, conversely, that the functional

requirements are complete. Table 2-1, the functional requirement/process map, shows which processes (in the functional model) satisfy each functional requirement. Table 2-2, the process/functional requirement map, shows which functional requirements are satisfied by each process in the model. Model completeness is demonstrated when each requirement is paired with at least one process, and vice versa.

2.4 FUNCTIONAL SPECIFICATIONS

This section presents the functional specifications for AODS. These specifications include computational models, algorithms, and procedures to be used in satisfying the functional requirements. There is not a one-to-one correspondence between the specifications and the requirements. However, the specifications have been grouped into the same functional areas as the functional requirements in Section 2.3.1. In cases in which a mathematical model is specified, the reference in which the algorithm can be found is cited rather than including the mathematics here. Those computational models that have not been specified to date are marked with the letters TBD, for "to be determined." These models will be specified through the GSFC Assistant Technical Representatives for the Microprocessor Software Support task.

2.4.1 SYSTEM CONTROL FUNCTIONAL SPECIFICATIONS

The functional specifications for system control are as follows:

- Perform the following functions in order of priority (the functions are listed according to their priority, with the highest priority first):
 - Capture input messages.
 - Service control commands as indicated in Table 2-3.

Table 2-1. Functional Requirement/Process Map

REQUIREMENT NUMBER	PROCESS NUMBER	REQUIREMENT NUMBER	PROCESS NUMBER
R1	P1	R3.6	P3.3
R1.1	P1.2, P1.3, P1.4	R3.7	P3.3
R1.2	P1.1, P1.4, P1.5	R4	P4
R1.3	P1.3	R4.1	P4.1, P4.2, P4.3
R1.4	P1.3, P1.4, P1.5, P2.3, P5.1	R4.2	P4.2.1, P4.2.2, P4.2.3
R1.4.1	P1.3, P3.3	R4.3	P4.1.2, P4.1.1, P4.1.3, P4.1.4, P4.1.5
R1.4.2	P1.3, P5.1.4	R4.4	P4.3
R1.5	P1.3, P1.4, P1.5	R5	P5
R1.6	P1.2, P1.3, P1.4, P1.5	R5.1	P5.2, P5.3, P5.4, P5.5, P5.6, P5.2.1, P5.2.2
R1.6.1	P1.2, P1.4, P1.5	R5.2	P5.2, P5.3, P5.2.1, P5.2.2
R1.6.2	P1.4, P1.5	R5.3	P5.1.2, P5.1.3
R1.6.3	P1.3, P1.4, P1.5	R5.4	P5.1
R1.6.4	P1.1, P1.4, P1.5	R5.4.1	P5.1.1
R1.6.5	P1.3, P1.4, P1.5	R5.4.2	P5.1.2, P5.1.3
R1.7	P1.3, P1.4, P1.5	R5.5	P5.1.1, P5.1.2, P5.1.3
R1.8	P1.2, P1.4, P1.5	R5.6	P5.2.3
R1.9	P1.2, P1.4, P1.5	R5.6.1	P5.2.3
R1.10	P1.2, P1.4, P1.5	R5.7	P5.2.4
R1.11	P1.2, P1.3, P1.4, P1.5	R5.8	P5.5
R1.12	P1.2, P1.4, P1.5	R5.8.1	P5.5
R1.13	P1.2, P1.4, P1.5	R5.8.2	P5.5
P1.14	P1.4, P1.5	R5.9	P5.6.2, P5.5
R2	P2	R5.10	P5.6.1, P5.5
R2.1	P2.1	R6	P6
R2.2	P2.2	R6.1	P6.1, P6.2, P6.3, P6.4
R2.3	P2.2, P2.4	R6.2	P6.2
R2.4	P2.1	R6.3	P6.1
R2.5	P2.2	R6.4	P6.3
R2.6	P2.3	R6.5	P6.2
R2.7	P2.5	R7	P7
R3	P3	R7.1	P7.4, P7.3, P7.5
R3.1	P3.1	R7.2	P7.1, P7.5
R3.2	P3.2	R7.3	P7.2, P7.5
R3.3	P3.1, P3.2	R7.4	P7.3, P7.5
R3.4	P3.3, P3.4, P4.1.1 P4.1.2	R7.5	P7.2, P7.5
R3.5	P3.3	R7.6	P7.2, P7.5

7359/80

Table 2-2. Process/Functional Requirement Map

PROCESS NUMBER	REQUIREMENT NUMBER	PROCESS NUMBER	REQUIREMENT NUMBER
P1	R1	P4.2	R4.1
P1.1	R1.2, R1.6.4	P4.2.1	R4.2
P1.2	R1.1, R1.6, R1.6.1, R1.8, R1.9, R1.10, R1.11, R1.12, R1.13	P4.2.2	R4.2
P1.3	R1.1, R1.3, R1.4, R1.4.1, R1.4.2, R1.5, R1.6, R1.6.3, R1.6.5, R1.7, R1.11	P4.2.3	R4.2
P1.4	R1.1, R1.2, R1.4, R1.5, R1.6, R1.6.1, R1.6.2, R1.6.3, R1.6.4, R1.6.5, R1.7, R1.8, R1.9, R1.10, R1.11, R1.12, R1.13, R1.14	P4.3	R4.1, R4.4
P1.5	R1.2, R1.4, R1.5, R1.6, R1.6.1, R1.6.2, R1.6.3, R1.6.4, R1.6.5, R1.7, R1.8, R1.9, R1.10, R1.11, R1.12, R1.13, R1.14	P5	R5
P2	R2	P5.1	R1.4, R5.4
P2.1	R2.1, R2.4	P5.1.1	R5.4.1
P2.2	R2.2, R2.3, R2.5	P5.1.2	R5.3, R5.4.2
P2.3	R1.4, R2.6	P5.1.3	R5.3, R5.4.2
P2.4	R2.3	P5.1.4	R1.4.2
P2.5	R2.7	P5.2	R5.1, R5.2
P3	R3	P5.2.1	R5.1, R5.2
P3.1	R3.1, R3.3	P5.2.2	R5.1, R5.2
P3.2	R3.2, R3.3	P5.2.3	R5.6, R5.6.1
P3.3	R1.4.1, R3.4, R3.5, R3.6, R3.7	P5.2.4	R5.7
P3.4	R3.4	P5.3	R5.1, R5.2
P4	R4	P5.4	R5.1
P4.1	R4.1	P5.5	R5.1, R5.8, R5.8.1, R5.8.2, R5.9
P4.1.1	R3.4, R4.1, R4.3	P5.6	R5.1
P4.1.2	R3.4, R4.3	P5.6.1	R5.10
P4.1.3	R4.3	P5.6.2	R5.9
P4.1.4	R4.3	P6	R6
P4.1.5	R4.3	P6.1	R6.1, R6.3
		P6.2	R6.1, R6.2, R6.5
		P6.3	R6.1, R6.4
		P6.4	R6.1
		P7	R7
		P7.1	R7.2
		P7.2	R7.3, R7.5, R7.6
		P7.3	R7.1, R7.4
		P7.4	R7.1
		P7.5	R7.1, R7.2, R7.3, R7.4, R7.5, R7.6

7359/80

Table 2-3. Service Control Commands and Results

ACTION COMMAND	OVERALL ACTION	INPUT	OUTPUT	COMPUTATIONS	DATA	CODE
REBOOT	CONTROL GIVEN TO BOOT ROUTINE	LOST	LOST	LOST	LOST	LOST
ABORT	ADDS ABORTED	ONLY START COMMAND ACCEPTED	ACTIVITY LOG DOWN LINKED	STOPPED ABRUPTLY	LOST	KEPT
STOP	DATA PROCESSING TERMINATED	ONLY START COMMAND ACCEPTED	OUTPUT COMPLETED NORMALLY	COMPLETED NORMALLY	LOST	KEPT
START	DATA PROCESSING INI TIATED	ALL DATA TYPES ACCEPTED	STANDARD OUTPUT INITIATED	REGULAR PROCESSING INITIATED	COLLECTED	NOT AFFECTED
SUSPEND	CURRENT COMPUTA TIONAL PROCESS SUSPENDED	ALL DATA TYPES ACCEPTED	STANDARD OUTPUT CONTINUES	SUSPENDED	KEPT	KEPT
CONTINUE	SUSPENDED COMPUTA TIONAL PROCESS RES UMED	ALL DATA TYPES ACCEPTED	STANDARD OUTPUT CONTINUES	RESUMED	KEPT	KEPT
STATUS REQUEST	ACTIVITY LOG OB TAINED	NOT AFFECTED	ACTIVITY LOG DOWN LINKED	NOT AFFECTED	NOT AFFECTED	NOT AFFECTED
SET CLOCK	CLOCK ADJUSTED FOR FAST TIMING	NOT AFFECTED	ADJUSTED FOR TIME CHANGE	ADJUSTED FOR TIME CHANGE	NOT AFFECTED	NOT AFFECTED

7359/50

- Output severe error messages.
- Perform the next time-scheduled function of the following: predict one-way Doppler observations; output one-way Doppler observations; predict and output state vector table; perform maneuver recovery; output activity log.
- Process input messages.
- Update TDRS files with new TDRS vector.
- Preprocess data.
- Perform estimation.
- Notify ground control of excessive idle time.
- Enter every system status message received from the other processes, every command received from ground control, and every directive generated into the activity log.
- Never perform data preprocessing if estimation is in progress.

2.4.2 INPUT PROCESSING FUNCTIONAL SPECIFICATIONS

The functional specifications for input processing are as follows:

- Identify and decode all input messages according to the input message formats specified in Appendix C.
- Accept as input a partition of executable code and load it into RAM. (The method is TBD.)
- Extract the usable bytes of each raw observation message and change binary numbers to real numbers where necessary (see Section C.1.2 in Appendix C).

- Queue the observations in the following manner:
 - Queue each observation type (one-way, two-way, and so forth) separately.
 - Within each observation type, queue the observations as they are received (in ascending time order).

2.4.3 DATA PREPROCESSING FUNCTIONAL SPECIFICATIONS

The functional specifications for data preprocessing are as follows:

- Check each observation for the proper tracking configuration by testing the following parts of the input observation:
 - Tracker type (TDRSS or SRE)
 - User support identification code (SIC)
 - Vehicle identification code (VID)
 - Ground antenna IDs (forward and return)
 - TDRS IDs (forward and return)
- Check each observation further for acceptability by testing the following:
 - Time tag
 - Valid/invalid switches for range and Doppler
- Make the following computational modifications to each acceptable observation:
 - Convert the time tag from year, seconds of year, and microseconds of second to a modified Julian date in universal time clock (UTC) time.
 - Modify the TDRSS range and Doppler measurements using the raw data reduction algorithms given in Reference 2.

- Compute the SRE range and range-rate measurements using the raw data reduction algorithms given in Reference 3.
- Propagate the TDRS using the following:
 - Integrator: Eighth-order predictor/corrector with fourth-order Runge-Kutta starter (Reference 4)
 - Method: Cowell fixed-step (Reference 4)
 - Force models for TDRS state:
 - Up to 15-by-15 geopotential field (Reference 4)
 - Solar radiation pressure (Reference 4)
 - Solar and lunar gravitational effects (analytic) (TBD)
 - Central body: Earth (Reference 4)
 - Special feature: Integration will be performed over full steps only. The propagator will interpolate for points that fall between steps. (The method of interpolation is TBD.)
- When a new TDRS vector is received, update the TDRS orbit file by propagating the new vector over the time span covered by the current file and replacing the records with the new vectors at the same time tags.
- When a TDRS maneuver occurs, update the TDRS integration start data by propagating the predicted TDRS state vector to the time tag of the TDRS integration start data and replacing it.

2.4.4 DATA MANAGEMENT FUNCTIONAL SPECIFICATIONS

The functional specifications for data management are as follows:

- Manage one observation file and three TDRS orbit files in memory.
- Locate, read, and write observation records in a sequential manner.
- Write TDRS orbit records in a sequential manner.
- Read TDRS records keyed by time.
- Update (i.e., read and then replace) TDRS records in a sequential manner.
- Create and manage all files such that records may be deleted and added without changing the size of the files.

2.4.5 ESTIMATION FUNCTIONAL SPECIFICATIONS

The functional specifications for estimation are as follows:

- Perform differential correction on the most recent batch of observation data (fixed-length time span) using the following:
 - Method: Batch least-squares (Reference 4)
 - Procedure: Sliding batch estimator--Each time a new pass of data is received, perform the following:

Move the epoch to the end time of the new pass.

Determine the start time of the estimation span by counting back the specified amount of time from epoch.

Process the observation data in reverse order (descending time).

Update the state at epoch at the end of each iteration.

Test for convergence at the end of each iteration. (The method is TBD.)

Generate DC Residuals Reports and DC Summary and Statistics Reports as specified by control parameters.

Stop estimation when convergence, divergence, or the maximum iteration is reached.

- Restrictions:

Do not begin estimation until the minimum time span of data has been received.

Do not perform estimation whenever the observation data span is less than the estimation time span.

● Propagate the user spacecraft (target) state and state transition matrix using the following:

- Integrator: Eighth-order predictor/corrector with a fourth-order Runge-Kutta starter (Reference 4)

- Method: Cowell fixed-step

- Force models for spacecraft state:

Up to 15-by-15 geopotential model (Reference 4)

Atmospheric drag (Harris-Priester) (Reference 4)

Solar and lunar gravitational effects
(analytic) (TBD)

Central body: Earth (Reference 4)

- Force models for the state transition matrix (10 solve-for parameters): J_2 , J_3 , J_4 , and drag (integrating variational equations) (method of integration is TBD)
- Special feature: Integration will be performed over full steps only. The propagator will interpolate for points that fall between steps. (The method of interpolation is TBD.)
- Retrieve the TDRS vectors required and interpolate to obtain the TDRS vector at the observation time.
- Perform n sigma editing (Reference 4).
- Check for a new initialization table and/or estimation control parameters during initialization and after suspension (by ground control) of a batch estimation process.
- Solve for the following state parameters:
 - Six parameters of the user spacecraft orbital state
 - One atmospheric drag coefficient, C_D (optional)
 - Three coefficients of the frequency model for one-way TDRSS data (optional)
- Model the following types of observations:
 - One-way TDRSS range and Doppler (Reference 3)
 - Two-way TDRSS range and Doppler (References 3 and 5)
 - Three-way TDRSS range and Doppler (Reference 3)

- One-way SRE range and range-rate (Reference 4)
- Two-way SRE range and range-rate (Reference 4)
- Resolve range ambiguity during observation modeling (Reference 5).
- Correct for ionospheric refraction in the observation model on option. (The method is TBD.)
- Compute the partial derivatives of the observations with respect to the following parameters:
 - User spacecraft orbital state (References 4 and 5)
 - Atmospheric drag (References 3 and 4)
 - Coefficients of frequency model for one-way TDRSS observations (Reference 3)

2.4.6 ONE-WAY DOPPLER FUNCTIONAL SPECIFICATIONS

The functional specifications for one-way Doppler prediction are as follows:

- Predict one-way Doppler observations over the input tracking interval using the specified TDRS.
- Retrieve the TDRS vector from the TDRS orbit file initially and then propagate it using the models given in the data preprocessing specifications (Section 2.4.3).
- Propagate the most recent state solution to obtain the user spacecraft state vector. If a solution has not been reached after a new initialization table has been input, the a priori state (in the initialization table) will be used as the reference vector for user spacecraft propagation. The propagation models given in the estimation specifications (Section 2.4.5) will be used.

- Do not check the scheduled observations for feasibility.

2.4.7 OUTPUT PROCESSING FUNCTIONAL SPECIFICATIONS

The functional specifications for output processing are as follows:

- Output all information in the output message formats specified in Appendix C.
- Generate a state vector predict table for the user spacecraft over a specified time interval, i , at a specified vector frequency, m minutes, as follows:
 - Each time a new solution is reached or a new a priori state vector (initialization table) is received, a table containing state vectors at m -minute intervals starting at the current time (t_n) and ending two time intervals later ($t_n + 2i$) will be generated and output.
 - Then, one time interval later ($t_n + i$), the next table will be generated and output. This table will contain state vectors at m -minute intervals over the next time interval. The start time of this table will be the end time of the previous table ($t_n + 2i$), and the end time will be one time interval later ($t_n + 3i$).
 - The second step will be repeated until a new solution is reached or a new a priori state vector is received, which causes the process to begin again with the first step.
- Propagate the user spacecraft using the models given in the estimation specifications (Section 2.4.5).

SECTION 3 - REQUIREMENTS ANALYSIS

This section presents the analysis of the AODS requirements. The AODS functional requirements were analyzed to determine the feasibility of implementing the system on an LSI-11/23 computer. Both the computer memory requirements and the AODS execution time requirements were determined. The timing requirements, which were defined through an extensive timing study, were applied to the operational scenario in a throughput analysis to show the feasibility of performing a worst case scenario within the specified time. In addition, an error contingency plan was developed for the operational scenario.

Section 3.1 presents the operational scenario. Sections 3.2, 3.3, and 3.4 present the timing and computer memory studies, the throughput analysis, and the error contingency plan, respectively.

3.1 OPERATIONAL SCENARIO

This section presents the standard internal operational scenario for AODS, which includes startup operations, observation data processing, and other processing.

The startup operational scenario is as follows:

1. The start command is uplinked.
2. The following must be uplinked before the first pass of tracking data:
 - Miscellaneous constants (optional)
 - Station constants (optional)
 - New TDRS vectors
3. The initialization table must be uplinked to initiate the generation and output of state vector predict tables.

4. The following must be uplinked to begin one-way Doppler prediction:
 - Initialization table
 - Tracking schedule
 - New TDRS vectors
5. The maneuver schedule must be uplinked before a maneuver will be scheduled.
6. The following must be uplinked before estimation will begin:
 - Initialization table
 - Estimation control parameters
 - Raw tracking data
7. The following data sets may be uplinked when necessary throughout execution:
 - Initialization table
 - Estimation control parameters
 - Maneuver schedule
 - Tracking schedule
 - New TDRS vectors
 - Raw tracking data
8. The following data sets may be uplinked when estimation is not in progress:
 - Station constants
 - Geopotential table
 - Atmospheric density table
 - Timing coefficients
9. A pass of tracking data is uplinked according to one of the following scenarios:
 - One burst of data containing one observation at the smallest interval

- One burst of data containing all observations covering a specified time interval

10. Data uplinking continues until the minimum estimation span is filled.

When a complete batch of data (specified length of time) has been received, AODS begins a standard operational procedure for each successive batch of data. This observation data processing scenario is as follows:

1. The new pass of observation data is captured and queued for the data preprocessor.
2. The new pass of data is preprocessed and written to the observation file. The TDRS vectors are propagated from the end of the previous pass to the end of the current pass of data and written to the TDRS orbit files.
3. Differential correction (DC) is performed on the next batch of data, which is determined by starting at the end time of the observation file (epoch) and counting back the specified amount of time. If not enough data exist to cover the estimation span, DC will not be performed.
4. DC is controlled by the estimation control parameters. DC reports are generated and output during estimation.
5. When a solution is reached, a new state vector predict table covering 1 hour in the future is generated and output.
6. If a solution is not reached by the maximum iteration of the DC, estimation stops until the next pass of data is received.

During the processing of observation data, other activities are being performed. The scenario for other processing is as follows:

1. One-way Doppler observations are predicted based on the most recent state solution. Prediction is scheduled so that the observations will be ready for output at the specified pad time before the start time of the current tracking interval.
2. State vector predict tables of a user-specified length continue to be generated and output at a user-specified frequency.
3. The activity log continues to be output at a specified time interval.
4. Priority (error) messages are output upon demand.
5. Commands are serviced immediately after reception.
6. All incoming data messages are captured and queued.
7. The following types of input are processed:
 - Estimation control parameters
 - Initialization table
 - Tracking schedule
 - Maneuver schedule
 - New TDRS vectors
 - Observation data
 - Commands

Other operations cannot be performed while estimation is active. The operations that are performed after estimation is completed on the current batch of data are as follows:

1. The following types of input are processed:
 - Station constants
 - Geopotential tables

- Drag tables
 - Timing coefficients
2. The TDRS orbit file(s) are updated with the new TDRS vector(s).
 3. TDRS maneuver recovery is performed.
 4. User spacecraft maneuver recovery is performed.
 5. The next pass of observation data is preprocessed.

3.2 TIMING AND COMPUTER MEMORY STUDIES

The objective of the timing and computer memory studies is to provide a basis for determining the timing and memory requirements of AODS. This section presents both the content of, and the results obtained from, these studies.

3.2.1 TIMING STUDY

The four activities necessary to determine the timing requirements for AODS are as follows:

1. Obtain a comparison of the execution time of the PDP-11/70 and LSI-11/23 computers.
2. Determine the actual time for each type of instruction by running tests on the PDP-11/70.
3. Time the existing pieces of software, which are similar to those required by AODS.
4. Define a method for estimating execution time based on FORTRAN code.

The results of this study are used during throughput analysis to estimate the timing requirements of the main functions in AODS. Due to the assumptions that were made in this study, a confidence level cannot be stated at this time. However, after a benchmark test is run on STL's PDP-11/70 and LSI-11/23 computers, the true LSI-to-PDP ratio will be

known, and a level of confidence will be assigned to this study.

3.2.1.1 Comparison of PDP-11/70 and LSI-11/23 Timing

AODS will be executed on two computers: the PDP-11/70 and the LSI-11/23. Because it must be demonstrated that the LSI-11/23 can perform the necessary computations within the time constraints, a comparison of execution times of the two computers is necessary. To obtain this comparison, the book values of the instruction times on both machines were obtained. Table 3-1 presents the instruction timing from the processor handbooks (References 6 and 7). The table specifies the minimum, maximum, and typical instruction times for the PDP-11/70 and the minimum and maximum times for the LSI-11/23, all of which were taken from the handbooks.

However, no typical time was specified in the handbooks for the LSI-11/23, so the average time was computed and is used as the typical time for that computer. The ratio of the LSI to the PDP was also computed for each instruction type. The ratios are large due to the special floating point package (FP-11/C) in STL's PDP-11/70.

Using only floating point values, it is estimated that AODS will consist of 60 percent load-and-compare (L&C) instructions, 20 percent single-precision (SP) arithmetic instructions, and 20 percent double-precision (DP) arithmetic instructions. Thus, using the average ratio of each of the instruction groups, the following equation represents the ratio of LSI-to-PDP execution time.

$$\begin{aligned}\text{Ratio LSI:PDP} &= 60\% \text{ L\&C} + 20\% \text{ SP} + 20\% \text{ DP} \\ &= 0.6 (37.73) + 0.2 (49.27) + 0.2 (76.58) \\ &= 47.8\end{aligned}$$

This ratio is used throughout the remaining timing studies and the throughput analysis as the relationship of the LSI

Table 3-1. Comparison of PDP-11/70 and LSI-11/23 Instruction Times

INSTRUCTION ^a	PDP-11/70 (WITH FP-11/C) TIMES (μ s)			LSI-11/23 TIMES (μ s)			RATIO LSI:PDP
	MINIMUM	MAXIMUM	TYPICAL	MINIMUM	MAXIMUM	AVERAGE	
LOAD-AND-COMPARE							
SP LOAD	0.36	0.36	0.36	9.15	18.77	13.96	38.78
DP LOAD	0.36	0.36	0.36	11.55	22.08	16.82	46.72
SP COMPARE	0.54	1.08	0.81	20.25	29.57	24.91	30.75
DP COMPARE	0.54	1.08	0.81	22.05	34.08	28.07	34.65
SP ARITHMETIC							
SP ADD	0.9	2.52	0.95	37.05	96.77	66.91	70.43
SP SUBTRACT	0.9	1.98	1.13	37.95	97.67	67.81	60.00
SP MULTIPLY	1.8	3.44	2.52	79.95	112.09	96.01	38.42
SP DIVIDE	1.92	6.72	3.54	91.05	108.77	99.91	28.23
DP ARITHMETIC							
DP ADD	0.9	4.14	0.98	42.45	184.08	113.27	115.58
DP SUBTRACT	0.9	4.14	1.16	43.35	184.98	114.17	98.42
DP MULTIPLY	3.06	6.22	4.68	193.05	280.98	237.02	50.65
DP DIVIDE	3.12	14.4	6.0	239.25	259.98	249.62	41.60

^aALL INSTRUCTIONS PRESENTED HERE ARE FLOATING POINT INSTRUCTIONS. SP = SINGLE-PRECISION;
DP = DOUBLE-PRECISION.

7359/80

to the PDP. The conclusion drawn from this study is that the LSI is 45 to 50 times slower than STL's PDP-11/70.

3.2.1.2 Actual PDP-11/70 FORTRAN Operation Execution Times

To estimate the time required to execute a piece of FORTRAN code, the total time required to perform the arithmetic operations and trigonometric functions is needed. This was obtained by timing each of these operations on the PDP-11/70 (dedicated) as follows:

1. A utility program was developed to time an individual line of code. To make the time large enough to be noticeable in seconds, a loop was added to execute the statement 10,000 times.
2. The code was timed without the statement containing the operation in question. This provided the overhead time in the loop.
3. Each operation was put in the simple arithmetic statement and timed. After subtracting the overhead, the average test time was computed for each type of operation. This process was repeated for the trigonometric functions.

Table 3-2 specifies the results of this study, including the minimum, maximum, and typical times for both single- and double-precision operations and functions. The typical times specified are used to estimate the timing requirements of the AODS functions.

3.2.1.3 Timing of Existing Software

Existing pieces of software, similar to those that will be developed for AODS, were located and timed on the PDP-11/70. Table 3-3 presents the results of these timing tests. In each case, the target code was executed 100 to 1000 times and averaged to obtain a reasonable time

Table 3-2. PDP-11/70 Timing Test Results

OPERATION	TIME (ms)		
	MINIMUM	MAXIMUM	TYPICAL
SP ADD	0.00664	0.00896	0.008
SP SUBTRACT	0.00664	0.00896	0.008
SP MULTIPLY	0.00816	0.01054	0.010
SP DIVIDE	0.00816	0.01366	0.012
SP SINE	0.07773	0.12149	0.110
SP COSINE	0.08164	0.13945	0.122
SP TANGENT	0.179688	0.18008	0.180
SP ARCSIN	0.21836	0.23789	0.234
SP ARCCOS	0.22774	0.25664	0.248
SP ARCTAN	0.13164	0.15352	0.143
SP SQRT	0.05453	0.05664	0.056
DP ADD	0.0082	0.01018	0.009
DP SUBTRACT	0.00816	0.00976	0.009
DP MULTIPLY	0.01133	0.01445	0.013
DP DIVIDE	0.01133	0.02851	0.022
DP SINE	0.13156	0.21132	0.095
DP COSINE	0.13156	0.21992	0.200
DP TANGENT	0.28477	0.28516	0.285
DP ARCSIN	0.21524	0.29336	0.278
DP ARCCOS	0.22695	0.22852	0.228
DP ARCTAN	0.20352	0.26992	0.237
DP SQRT	0.07695	0.08633	0.083

7359/80

Table 3-3. Execution Times of Existing Software Components

TIMING TEST	COMPONENT	EXECUTION TIME
USER SPACECRAFT (SMM) PROPAGATION: 60,000-SECOND SPAN, 60-SECOND STEPSIZE (1000 STEPS), 2-BY-2, DRAG	INTEL PROPAGATOR	9.0 SECONDS (0.009 SECOND PER STEP)
USER SPACECRAFT (SMM) PROPAGATION: 60,000-SECOND SPAN, 60-SECOND STEPSIZE (1000 STEPS), 8-BY-8, DRAG	INTEL PROPAGATOR	15.7 SECONDS (0.016 SECOND PER STEP)
USER SPACECRAFT (SMM) PROPAGATION: 60,000-SECOND SPAN, 60-SECOND STEPSIZE (1000 STEPS), 8-BY-8, DRAG, SUN AND MOON	INTEL PROPAGATOR	28.4 SECONDS (0.028 SECOND PER STEP)
TDRS SPACECRAFT PROPAGATION: 60,000- SECOND SPAN, 60-SECOND STEPSIZE (1000 STEPS), 8-BY-8, SOLAR RADIATION, SUN AND MOON	INTEL PROPAGATOR	27.9 SECONDS (0.028 SECOND PER STEP)
10-BY-10 MATRIX INVERSION	SUBROUTINE IV	0.492 SECOND
TWO-WAY RANGE AND DOPPLER COMPUTA- TION WITHOUT PARTIALS	TWO-WAY OBSER- VATION MODEL	0.024 SECOND
TWO-WAY RANGE AND DOPPLER COMPUTA- TION WITH PARTIALS	TWO-WAY OBSER- VATION MODEL	0.029 SECOND

7359/80

estimate. These times are used in the throughput analysis and in the computation of program overhead time.

3.2.1.4 Estimation of Execution Time

Using the FORTRAN operation execution times determined in Section 3.2.1.2 and the FORTRAN code for a particular function, it should be possible to estimate the execution time required to perform that function. However, Table 3-2 does not contain times for operations such as load, store, branch, and compare or for integer arithmetic operations (index computations, counters, and so forth). In sample cases in which the actual execution time was known, the execution time was also estimated based on the FORTRAN code. It was found that on the average the estimate was approximately one-half the actual execution time. After examining the code, this difference was attributed to the previously specified operations for which actual execution times are not known. Thus, when estimating execution time using this method, the estimate should be increased by a factor of two to account for noncomputational statements and integer arithmetic.

3.2.1.5 Input/Output Timing

The time required to perform input or output was determined using the information given in Appendix C. First, the total size of each type of data block, including all record headers and filler bytes, was computed. Then the transmission times were computed based on two different transmission rates: a typical flight transmission rate (2100 bits per second for uplink and 56,200 bits per second for downlink) and the speed of the lines that will be used for data transmission in the prototype system (9600 bits per second for both directions). Tables 3-4 and 3-5 present the transmission times for input and output messages, respectively. These times are used in the throughput analysis.

Table 3-4. Input Message Transmission Times

INPUT MESSAGE BLOCK	TOTAL BYTES	UPLINK TIME (sec)	
		AT 2100 BITS/ SEC	AT 9600 BITS/ SEC
OBSERVATIONS (BURST OF 60)	5,120	19.5	4.267
OBSERVATIONS (60 BURSTS OF 1)	15,360	58.51	12.8
INITIALIZATION TABLE	256	0.975	0.213
NEW TDRS VECTORS (3)	768	2.9	0.64
ESTIMATION CONTROL PARAMETERS	256	0.975	0.213
MANEUVER SCHEDULE	1,024	3.9	0.853
TRACKING SCHEDULE	512	1.95	0.4267
MISCELLANEOUS CONSTANTS	256	0.975	0.213
STATION CONSTANTS	1,280	4.876	1.067
GEOPOTENTIAL TABLES	1,280	4.876	1.067
ATMOSPHERIC DENSITY TABLES	768	2.9	0.64
TIMING COEFFICIENTS	256	0.975	0.213
COMMAND	256	0.975	0.213

7359/80

Table 3-5. Output Message Transmission Times

OUTPUT MESSAGE BLOCK	TOTAL BYTES	DOWNLINK TIME (sec)	
		AT 56,200 BITS/ SEC	AT 9,600 BITS/ SEC
STATE VECTOR PREDICT TABLE (1 HOUR)	2,816	0.05	0.293
STATE VECTOR PREDICT TABLE (½ HOUR)	1,536	0.027	0.16
PREDICTED ONE-WAY DOPPLER (100 OBSERVATIONS)	3,328	0.0592	0.3467
PRIORITY MESSAGE	256	0.00456	0.0267
ACTIVITY LOG	5,632	0.1	0.5867
DC SUMMARY AND STATISTICS REPORT	768	0.0137	0.08
DC RESIDUALS REPORT	26,112	0.465	2.72

7359/80

3.2.2 COMPUTER MEMORY REQUIREMENTS

The computer memory requirements for AODS were estimated based on similar existing software and the amount of data to be stored. The data storage requirements were computed assuming that all data must be stored internally in memory, since no peripherals will be available. In addition, the size of the software components was estimated with the understanding that overlaying will not be possible.

Table 3-6 specifies the AODS data storage requirements. The sizes of the observation and TDRS orbit files were estimated based on the data dictionary (Appendix B) and the functional requirements given in Section 2.3.1. The sizes of the other data storage were estimated according to the description of the data given in the data dictionary and the input and output message formats (Appendix C).

Table 3-7 presents the estimated sizes of the AODS software components. The components listed will perform the main functions in AODS. Since the sizes were estimated based on similar existing software components of other systems, the names of the system in which it was found and the computer on which it is implemented are also provided.

Based on the estimates presented here, AODS as it is defined in Section 2 will fit on an LSI-11/23 with 256K bytes of RAM. AODS will occupy approximately 200K bytes, and the operating system and handlers will use 12K bytes, leaving about 44K bytes for the input message queue (which is not included here because it can be made a variable size). The memory requirements specified here are only estimates and are subject to error. If, for example, the size estimates were under by 15 percent, there would probably be insufficient space for the program and data as presented here, and further studies would be required.

Table 3-6. AODS Data Storage Requirements

TYPE OF DATA	SIZE (bytes)
OBSERVATION FILE (500 OBSERVATIONS)	25,000
TDRS ORBIT FILES (50 HOURS)	28,944 (9,648 PER FILE)
INITIALIZATION TABLE	188
NEW TDRS VECTORS	174
ESTIMATION CONTROL PARAMETERS	62
MANEUVER SCHEDULE	928
TRACKING SCHEDULE	288
CONSTANTS	3,078
CONTROL COMMAND	100
STATE VECTOR PREDICT TABLE (1 HOUR'S WORTH OF VECTORS AT 30- SECOND INTERVALS)	1,260
PREDICTED ONE-WAY DOPPLER OBSERVATIONS (100 OBSERVATIONS)	2,904
ERROR MESSAGE	46
ACTIVITY LOG	5,000
DC SUMMARY AND STATISTICS REPORT	504
DC RESIDUALS REPORT (500 OBSERVATIONS)	24,000
BATCH MATRICES	880
STATE SOLUTION	88
OBSERVATION QUEUES	4,400
ESTIMATION PARAMETERS	400
TDRS INTEGRATION START DATA (180 PER PROPAGATION)	1,080
TARGET INTEGRATION START DATA (180 PER PROPAGATION)	540
TARGET STATE TRANSITION MATRIX INTEGRATION ACCELERATIONS	1,460
TOTAL ESTIMATED AODS DATA STORAGE REQUIREMENT:	101,324 (102K BYTES)

7359/80

Table 3-7. Estimated Size of AODS Software Components

AODS COMPONENT	SOURCE OF EXAMPLE	MACHINE	ESTIMATED SIZE (K-bytes)
ORBIT PROPAGATOR (SUN, MOON, DRAG, 8-BY-8)	INTEL PROPAGATOR	PDP-11/70	12
GEOPOTENTIAL MODEL (15-8Y-15)	GTDS	IBM S/360-95	4
STATE TRANSITION MATRIX	GTDS	IBM S/360-95	4
OBSERVATION MODELS			
TWO-WAY TDRSS	TWO-WAY TDRSS MODEL	PDP-11/70	5
ONE-WAY TDRSS	TWO-WAY TDRSS MODEL	PDP-11/70	5
THREE-WAY TDRSS	TWO-WAY TDRSS MODEL	PDP-11/70	5
ONE-WAY SRE	GTDS	IBM S/360-95	5
TWO-WAY SRE	GTDS	IBM S/360-95	5
BATCH WEIGHTED LEAST-SQUARES	ENTREE	CDC CYBER 170	2
MATRIX MULTPLICATION AND INVERSION	ENTREE	CDC CYBER 170	2
OBSERVATION ACCEPTANCE AND REDUCTION	GTDS	IBM S/360-95	2
ESTIMATION EXECUTIVE	ENTREE	CDC CYBER 170	6
INPUT PROCESSING			8
OUTPUT PREPARATION AND TRANSMISSION			4
DATA MANAGEMENT			4
OVERALL EXECUTIVE (CONTROL)			16
TOTAL ESTIMATED AODS SOFTWARE SIZE:			89 ± 10% (UPPER LIMIT: 98)

08/6/70

3.3 THROUGHPUT ANALYSIS

The object of throughput analysis is to show the feasibility of executing the proposed software on the specified hardware system in the required amount of time. This is done by identifying a worst case path through the system, estimating or locating the actual execution time for each of the major functions in that path, and computing the execution time of that path. Usually, two throughput analyses are performed. First, a worst case throughput analysis is performed to show the feasibility of implementing the software system on the specified hardware. Second, a typical case throughput analysis is performed to determine the typical amount of execution time that will be required to perform the intended functions and, conversely, the amount of idle time that will be available for other processing.

Due to the real-time processing requirement, AODS must be able to complete orbit determination on each batch of observation data before the next pass of raw tracking data is received. A pass of tracking data is to be uplinked once per user spacecraft revolution. In addition, AODS must perform all its regularly scheduled functions (e.g., one-way Doppler prediction and output, state vector predict table generation and output, activity log output, data capture) during the same processing period.

For the throughput analyses presented in this section, the maximum possible processing time is 90 minutes (5400 seconds), which is considered to be the minimum revolution time of a user spacecraft. Sections 3.3.1 and 3.3.2 present the throughput analyses of the worst case and the typical case, respectively. Each case covers the timespan from the time that a pass of observations is received until the next pass of data is received (1-1/2 hours later). The cases are

broken down into the major functions performed, and an estimated execution time is assigned to each one. The execution times were estimated in the following manner:

- All propagation times were estimated using the maximum execution time per step (0.028 second) specified in Table 3-3. It was assumed that the inclusion of the Sun and Moon forces in the propagation of the user spacecraft more than compensated for the lack of an actual time for the state transition matrix propagation (variational equations). (The largest geopotential table used was 8-by-8. If a 15-by-15 table were to be used, the time per step would increase significantly.)
- The matrix inversion and observation modeling times were taken from Table 3-3.
- The data capture and output times were taken from Tables 3-4 and 3-5, respectively. The data rate of 9600 bits per second, which will be used by the prototype AODS, was used.
- The data retrieval times were estimated based on a task interface time of 0.2 millisecond (see Reference 6).
- All other times were estimated using the operation times specified in Table 3-2 and the computational model for the function. These times have been adjusted by the overhead factor determined in Section 3.2.1.4.

The execution times in the throughput analyses are in units of seconds unless otherwise specified. Times marked with an asterisk (*) are input/output times.

3.3.1 WORST CASE THROUGHPUT ANALYSIS

The functions included in the worst case for processing one batch of data in AODS are as follows:

<u>AODS Function</u>	<u>PDP-11/70 Execution Time (Seconds)</u>
1. Capture one pass of raw tracking data (100 observation pairs)	7.112*
2. Preprocess 100 observation pairs	0.015
3. Pregenerate three TDRS orbit files over the next revolution (90-minute span, 10-minute stepsize)	0.756
4. Generate two 30-minute state vector tables (60-second stepsize)	1.68
5. Output two 30-minute state vector tables	0.32*
6. Initialize estimation (move epoch)	2.52
7. Perform one batch iteration	
a. Retrieve 500 observation pairs	0.1
b. Retrieve TDRS vectors for interpolation (groups of eight)	0.0052
c. Propagate user spacecraft state vector and state transition matrix over 12 hours (60-second stepsize)	20.16
d. Compute 500 observation pairs with partials	14.5
e. Propagate observation partials to epoch (1000 observations)	3.74
f. Accumulate batch statistics (1000 observations)	6.58
g. Invert normal matrix (10-by-10)	0.49
h. Solve for state update (10 parameters)	0.004
i. Update state vector	0.0002
j. Generate DC Summary and Statistics Report	0.002

AODS Function	PDP-11/70 Execution Time (Seconds)
k. Output DC Summary and Statistics Report	0.080
(Total for 7: 45.6614)	
8. Generate DC Residual Report	0.12
9. Output DC Residual Report	2.72*
10. Assuming convergence occurred, generate new 60-minute state vector predict table based on new solution	1.68
11. Output 60-minute state vector table	0.293*
12. Capture and store input data	
a. Capture the following data sets:	
(1) Maneuver schedule	0.853*
(2) Tracking schedule	0.427*
(3) Estimation control parameters	0.213*
(4) Initialization table	0.210*
(5) Station parameters	1.067*
(6) Geopotential tables	1.067*
(7) Drag tables	0.640*
(8) Timing coefficients	0.213*
b. Store these data sets	2.84
(Total for 12:	
13. Capture three new TDRS vectors	0.64*
14. Update three TDRS orbit files with new vectors (12-hour span, 10-minute stepsize)	6.048
15. Predict one-way Doppler observations (two 15-minute tracking intervals, one observation per 10 seconds)	5.16
16. Output one-way Doppler observations (two intervals)	0.624*
17. Output three activity logs	1.76*
18. Output priority message	0.027*

To demonstrate the feasibility and/or limitations of implementing AODS on the LSI-11/23, the number of batch iterations possible in the 90-minute maximum time span are

computed. First the input/output and central processing unit (CPU) times for functions that are not part of the batch iteration are summed:

<u>Function</u>	<u>CPU Time (Seconds)</u>	<u>Input/Output Time (Seconds)</u>
1		7.112
2	0.015	
3	0.756	
4	1.68	
5		0.32
6	2.52	
8	0.12	
9		2.72
10	1.68	
11		0.293
12	2.84	4.69
13		0.64
14	6.048	
15	5.16	
16		0.624
17		1.76
18		0.027
	<hr/>	<hr/>
	20.819	18.186

Using the worst case execution time for one batch iteration, the number of iterations possible can be computed as follows:

$$5400 = 20.819 + 18.186 + 45.66x$$

$$5400 = 39 + 45.66x$$

$$5361 = 45.66x$$

$$x = 117.4$$

Thus, 117.4 iterations could be performed on the PDP-11/70. Converting the CPU times to LSI-11/23 times, the total

number of iterations possible on the LSI-11/23 can be computed as follows:

$$\begin{aligned} 5400 &= 995.1482 + 18.186 + 2182.548x \\ 5400 &= 1013.33 + 2182.548x \\ 4386.67 &= 2182.548x \\ x &= 2.01 \end{aligned}$$

Thus, only 2.01 batch iterations could be performed on the LSI-11/23 in the worst case.

3.3.2 TYPICAL CASE THROUGHPUT ANALYSIS

The functions included in the typical case for processing one batch of data in AODS are as follows:

AODS Function	PDP-11/70 Execution Time (Seconds)
1. Capture one pass of raw tracking data (60 observation pairs)	4.267*
2. Preprocess 60 observation pairs	0.013
3. Pregenerate three TDRS orbit files over the next revolution (90-minute span, 10-minute stepsize)	0.756
4. Generate the 30-minute state vector tables (60-second stepsize)	1.68
5. Output two 30-minute state vector tables	0.32*
6. Initialize estimation (move epoch)	2.52
7. Perform one batch iteration	
a. Retrieve 480 observations	0.096
b. Retrieve TDRS vectors for inter- polation (groups of eight)	0.0052
c. Propagate user spacecraft state vector and state transition matrix over 12 hours (60-second stepsize)	20.16
d. Computer 480 observation pairs with partials	13.92
e. Propagate observation partials to epoch (960 observations)	3.59

<u>AODS Function</u>	<u>PDP-11/70 Execution Time (Seconds)</u>
f. Accumulate batch statistics (960 observations)	6.32
g. Invert normal matrix (10-by-10)	0.49
h. Solve for state update (10 parameters)	0.004
i. Update state vector (10 parameters)	0.0002
j. Generate DC Summary and Statistics Report	0.002
k. Output DC Summary and Statistics Report	0.08
(Total for 7: 44.6674)	
8. Generate DC Residual Report	0.12
9. Output DC Residual Report	2.72*
10. Assuming convergence occurred, generate new 60-minute state vector predict table based on new solution	1.68
11. Output 60-minute state vector table	0.293*
12. Predict one-way Doppler observations (one 10-minute tracking interval, one observation per 10 seconds)	1.72
13. Output one-way Doppler observations (one interval)	0.208*
14. Output three activity logs	1.76*
15. Output priority message	0.027*

All functions in the typical case that are not part of the batch iteration are summed as follows:

<u>Function</u>	<u>CPU Time (Seconds)</u>	<u>Input/Output Time (Seconds)</u>
1		4.267
2	0.013	
3	0.756	
4	1.68	
5		0.32
6	2.52	

<u>Function</u>	<u>CPU Time (Seconds)</u>	<u>Input/Output Time (Seconds)</u>
8	0.12	
9		2.72
10	1.68	
11		0.293
12	1.72	1.72
13		0.208
14		1.76
15		<u>0.027</u>
	<u>8.489</u>	9.595

Using the typical execution time for one batch iteration, the number of iterations possible on the PDP-11/70 can be computed as follows:

$$\begin{aligned}
 5400 &= 8.489 + 9.595 + 44.66x \\
 5400 &= 18.084 + 44.66x \\
 5381.9 &= 44.66x \\
 x &= 120.5
 \end{aligned}$$

Converting the CPU times into LSI-11/23 times, the typical number of iterations possible on the LSI-11/23 can be computed as follows:

$$\begin{aligned}
 5400 &= 405.77 + 9.595 + 2134.748x \\
 5400 &= 415.37 + 2134.748x \\
 4984.63 &= 2134.748x \\
 x &= 2.335
 \end{aligned}$$

Thus, in the typical case, only 2.335 batch iterations can be performed.

3.3.3 CONCLUSIONS AND RECOMMENDATIONS

The key to the timing problem is the orbit propagator. A careful study of the throughput analyses shows that 45 percent of each batch iteration is spent propagating the user

spacecraft state. In addition, approximately 60 to 80 percent (64 percent in the worst case and 79 percent in the typical case) of the CPU time of the other (noniterative) activities is spent in the orbit propagator. It is to be remembered that an 8-by-8 geopotential model and a fourth-order Runge-Kutta integrator were used in the the throughput analyses. If a larger model such as a 15-by-15 model were used, the time spent in the propagator would be longer. Therefore, great care should be taken in selecting the force models that will be implemented in AODS. That is, no more than those absolutely necessary should be included.

An alternative to the reduction of the orbit propagator to a "bare-bones" force model is the addition of a second micro-processor that would perform all orbit propagation, thereby removing it from the main processor. This would allow the main processor to request an orbit propagation and then proceed with its other functions while the second processor performs the propagation. Using the execution times specified in Section 3.3.2, the effectiveness of this configuration can be demonstrated. By reducing the execution times of the noniterative functions and the batch iteration by 64 and 45 percent, respectively, the following equation is obtained:

$$\begin{aligned}5400 &= 265.84 + 1174.11x \\5134.16 &= 1174.11x \\x &= 4.373\end{aligned}$$

Thus, the productivity of AODS has in effect been doubled.

Another alternative is to change the method of estimation to a noniterative method such as a standard or extended Kalman filter. Since the time required to perform the estimation by filtering would be approximately the same as that required to perform one batch iteration, even the worst case would have a substantial amount of "free time." However,

the question of estimation reliability must dictate the feasibility of this suggestion.

3.4 ERROR CONTINGENCY PLAN

This section, which presents a philosophy for error handling in AODS, is intended to advise the designer in handling those errors which will be identified during AODS design. AODS errors fall into four general categories: computational errors, algorithmic errors, input data errors, and operational errors. All errors will be recorded on the activities log, and severe error messages will be downlinked immediately. The following subsections give the philosophy for handling each of the error types and some examples of each.

3.4.1 COMPUTATIONAL ERRORS

Computational errors are arithmetic errors that are detected by the machine (e.g., overflow, underflow, divide check). These errors should be avoided by implementing safeguards in the AODS code as well as removing outlying data points that would randomly cause such errors.

3.4.2 ALGORITHMIC ERRORS

Algorithmic errors, which will be detected by AODS itself, are those errors that occur as part of, or as a result of, a particular algorithm or computational model (e.g., estimator divergence, normal matrix singularity, editing of too much data by the estimator). These errors will be handled according to the specifications that will be given by the associated GSFC analysts. However, until that time, a temporary procedure for handling these errors has been defined. In cases of algorithmic errors in estimation, the estimator will be terminated, and the state vector will not be updated.

3.4.3 INPUT DATA ERRORS

There are two types of input data errors: transmission errors and numerical errors. In the case of transmission errors (e.g., incomplete block, missing records, unidentified data type), the data block should be discarded and a re-transmission request downlinked immediately.¹ In the case of numerical errors (e.g., unreasonable values for observation measurements and time tags), the data should be discarded and the errors recorded on the activity log.

3.4.4 OPERATIONAL ERRORS

Operational errors are those errors which could occur as a result of the operation of the program and/or would have a significant effect on program operation. Because of the wide variety of errors in this category, each error requires special error handling. Three AODS errors of this type are foreseen at this time. These operational errors and their solutions are as follows:

ERROR: AODS falls behind schedule in processing.

SOLUTION: Since estimation will be the lowest priority function, it will absorb the effects of this problem. The estimator will continue until it has completed estimation on its current batch of data. When it has completed estimation, the next pass of data will be added to the observation file. However, no estimation will occur on this batch of data. When the next pass of data is received, it will be added to the observation file. The estimator will then slide forward (two passes of data), and normal processing will resume. By eliminating one estimation process,

¹It is assumed that the normal system hardware will perform this function.

much time will be saved without seriously impacting the regular operation of AODS.

ERROR: The maneuver or tracking schedule is received late (i.e., received after the scheduled start time).

SOLUTION: AODS should assume that all times earlier than the "current time" are in error. AODS will record the errors and resume normal operations with the first entry that is greater than or equal to the current time.

ERROR: Constants (station constants, geopotential tables, drag tables, timing coefficients) are received during estimation.

SOLUTION: AODS will ignore (discard) these data and record the error in the activity log.

SECTION 4 - AODS DEVELOPMENT STRATEGY

This section, intended as a guide for the AODS designer, presents a philosophy of implementation, or development strategy, for AODS.

To achieve a prototype AODS by the spring of 1981, it will be necessary to implement the system in steps, each step demonstrating a major feature of the final system. Figure 4-1 shows this proposed systematic development from the early microprocessor activities (i.e., development of the IMP-16 microprocessor Onboard Determination System (ODS) demonstration system; see Reference 8) to the final flight-qualified system. This implementation plan is based on the time schedule specified by GSFC and has not been modified based on this document. The major milestones of the proposed schedule are as follows:

- Current activities
 - Design the prototype AODS from the results of the requirements analysis.
 - Implement portions of the system such that the primary input and output capabilities can be fully exercised and checked in a simulation environment.
 - Design and implement the environment simulator, ADEPT.
- Future activities
 - Add to the prototype AODS the mathematical algorithms specified in (1) the previous design activity and (2) analysis performed outside the microprocessor Software Support task.
 - Move the system to, and test it on, the LSI-11/23 microcomputer.

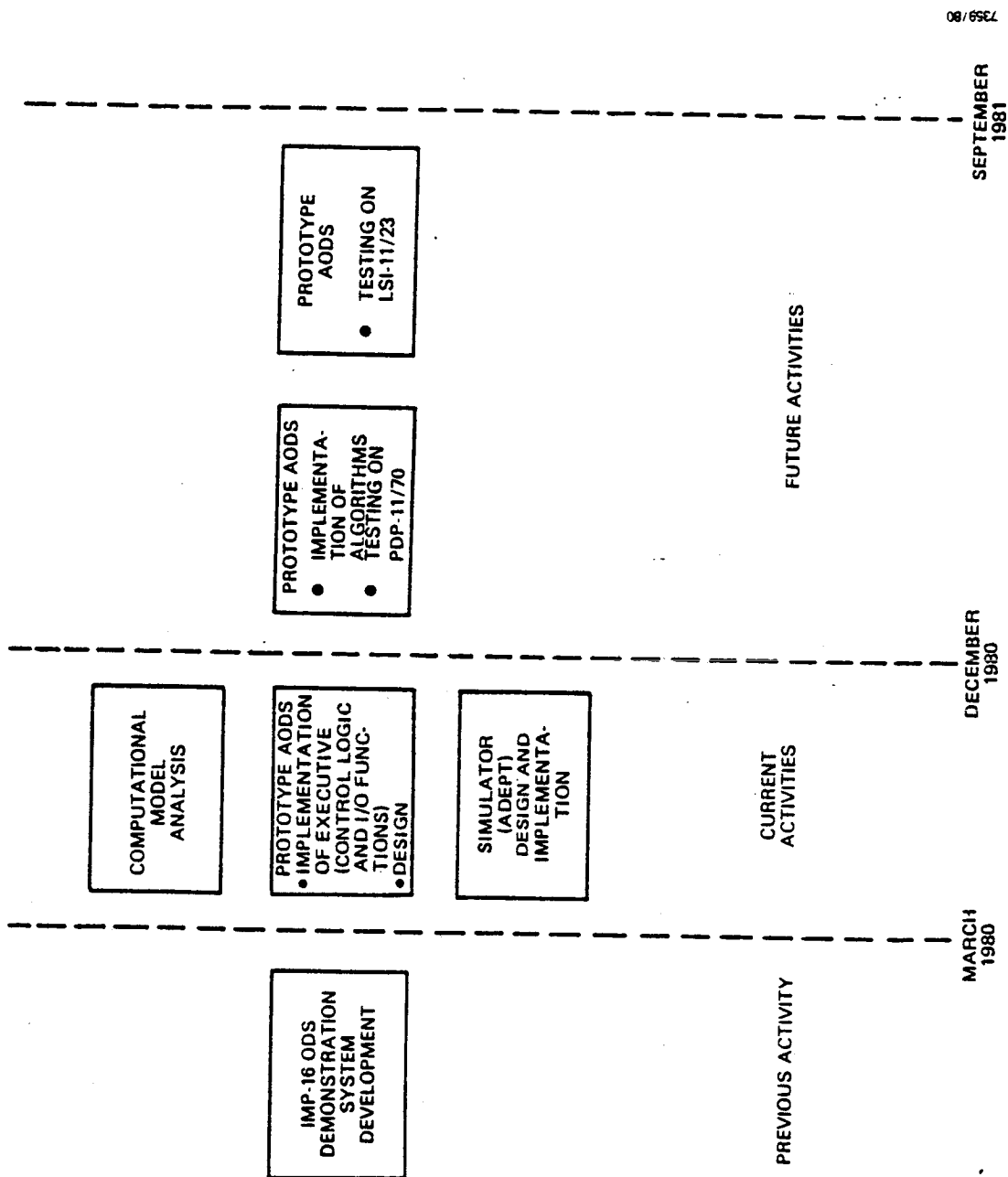


Figure 4-1. GSFC-Specified Schedule for AODS Development

This section discusses the implementation of the system according to the proposed time schedule. Section 4.1 discusses the implementation of the prototype AODS executive and simulator (first build) and the assembly of the prototype system on the PDP-11/70 minicomputer. Section 4.2 discusses the implementation of the mathematical algorithms into the prototype system; the moving of the system to, and its testing on, the LSI-11/23 microcomputer (second build); and the development and implementation of a final flight-qualified system. Throughout the development of both AODS and ADEPT, modern software practices will be adhered to; specifically, these include use of the following:

- Top-down design
- Design reviews
- Code walk-throughs
- Librarian
- Chief programmer

4.1 CURRENT ACTIVITIES (BUILD 1)

The current statement of work (Task 989, Amendment 2) describes the goals to be achieved during the current tasking period (spring 1980 through fall 1980). The statement directs that the AODS executive (driver) will be implemented and tested on the STL PDP-11/70 minicomputer. The degree of implementation will be such that primary input and output capabilities can be fully exercised and checked in a simulation environment. In addition, the control logic for invoking various subsystems and the corresponding stubs will be implemented. The environment simulator, ADEPT, will also be fully implemented and tested. Sections 4.1.1 and 4.1.2 discuss the AODS and ADEPT implementation strategies, respectively, and the assembly of the prototype system on the PDP-11/70.

4.1.1 AODS IMPLEMENTATION (FIRST BUILD)

Although AODS will be completely designed under the current tasking period (spring 1980 through fall 1980), only selected system processes will be implemented. The processes to be implemented include

- Data capture
- Input message processing
- Executive control
- Output message preparation
- Output message transmission

The remaining processes will be implemented as stubs with their proper interfaces. The following processes will not be implemented:

- Data preprocessing
- State estimation
- Data base management (observations and TDRS vectors)
- One-way Doppler observation prediction
- State vector predict table generation

Implementation will begin when the design of the processes to be implemented has been completed. During this first-build implementation stage, the design of the remaining processes (i.e., those not to be implemented at this stage) will be completed.

4.1.1.1 Data Capture

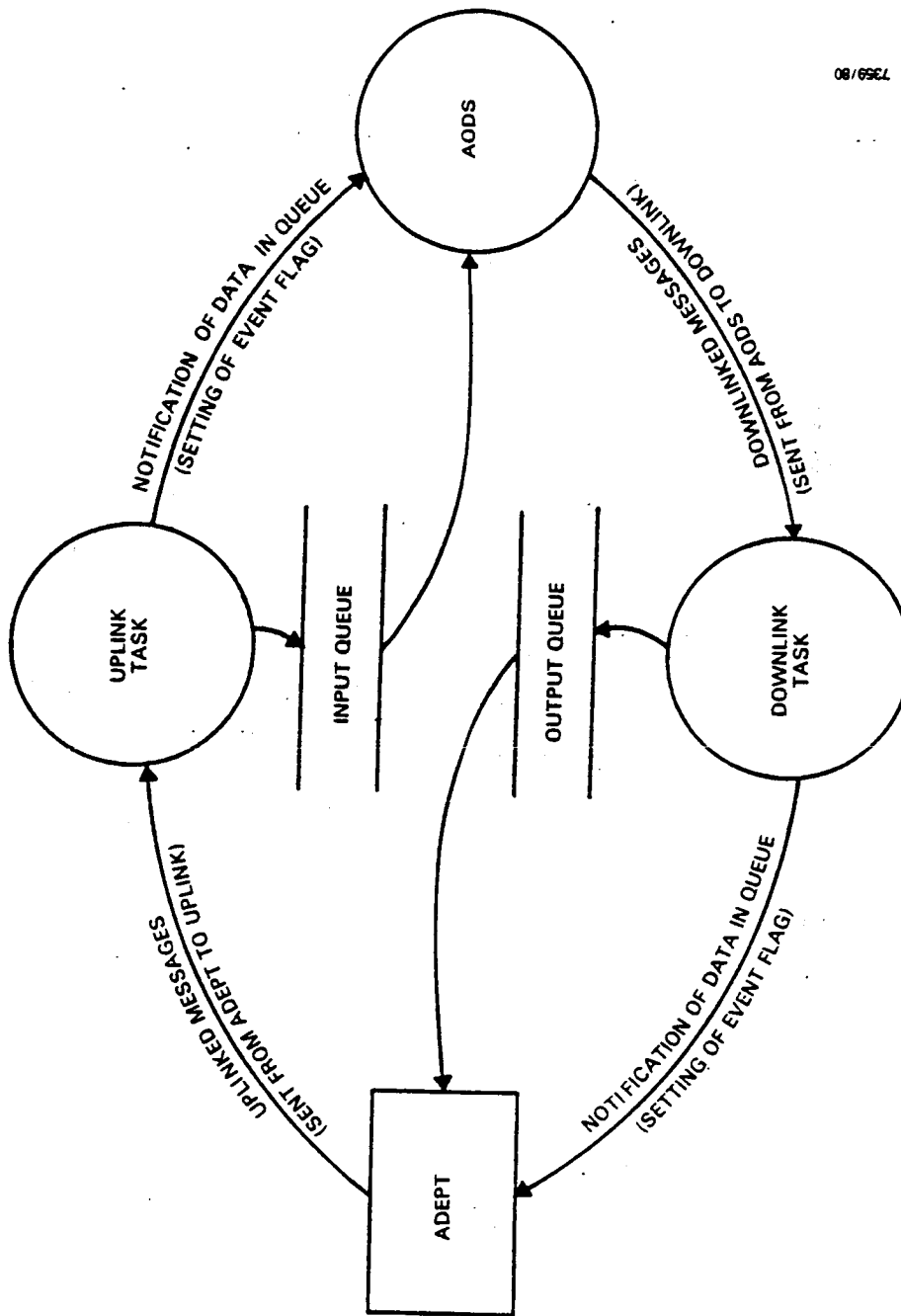
AODS will capture all messages uplinked to AODS from ADEPT and put these messages in an input queue for use by the input message processor. In the first build, both AODS and ADEPT will reside in the same PDP-11/70 computer, and communications over an external line will not be necessary to

pass data between programs.¹ Instead, a scheme such as that shown in Figure 4-2 will be used. Figure 4-2 shows two tasks (uplink and downlink) that act as the communications interface between ADEPT and AODS. Each task, when given control, loads data in the appropriate queues in the same manner as will the final communications software. The two tasks will have a high priority in the system and should be serviced immediately. An additional function of the uplink task will be to recognize commands and, in many cases, initiate the action to carry out the commands. The uplink tasks will also notify the AODS executive if the input queue is nearly full. This scheme should protect AODS and ADEPT from major modifications when, in future builds, AODS is moved to the LSI-11/23, and external communications (i.e., QIO and WTQIO system directives using an asynchronous trap (AST)) are required. The first-build data capture routine will not allow loading of partitions of code or perform any validity checking (such as parity or cyclic redundancy).

4.1.1.2 Input Message Processing

The first-build AODS will perform all functions specified in the AODS input requirements, including decoding and identifying all input messages, storing the identified data in the proper data sets, and extracting the usable portions of the raw observation message and storing the reduced observation in the proper queue based on observation type. Specifically, the first-build input message processor will accept

¹Work has been done to simulate communications between the PDP-11/70 and the LSI-11/23 by installing wraparound cabling between input/output ports on the PDP-11/70. The current work is preliminary, and techniques to communicate over the lines have not been finalized.



7359/80

Figure 4-2. AOPS/ADEPT First-Build Internal Communications Design

data from the data capture input message queue and more data into the proper COMMON block. The data to be moved include

- Raw observation messages
- New TDRS vectors
- Maneuver schedule
- Initialization tables
- Tracking schedule
- Miscellaneous constants
- Estimation control parameters
- Station parameters
- Geopotential tables
- Atmospheric drag tables
- Timing coefficients

The receipt of all input messages will be recorded in a COMMON block used by the executive to form the activities log. The input message processor will be called from the executive only when there is an indication of data in the input queue. If there is an unusual condition in the input message processor (e.g., full reduced observation queue, missing records, incomplete records, mixing of different types of data in a block, error in data size in a record, invalid type of data), the task will report this condition to the executive.

4.1.1.3 Executive Control

Executive control is the key part of AODS implementation. Design decisions made relative to the AODS executive will affect all tasks in the system, thus much attention will be given to this important software module and in the first-build AODS.

The AODS executive will determine scheduling needs (next action(s)) through a decision tree or table; maintain a table of active tasks and their levels of priority; communicate with the RSX-11M operating system coordinating the

tasks to be scheduled and their levels of priority; specify the amount of time (i.e., time slice) the scheduled tasks are allowed for completion before control returns to the executive; and maintain and output the activities log. Figure 4-3 shows the basic scheduling mechanism of the executive, which is as follows:

1. Through input (e.g., tracking schedule, maneuver schedule, status of task currently scheduled, command messages, system clock time), the executive determines the content of the scheduling table.
2. The executive reads the scheduling table, resuming tasks in the requested order (execution priority).
3. Using a mark time system directive and an AST, the executive sets up a time at which to be reactivated.

An AODS executive design goal will be to design as much of the software as possible in FORTRAN, but in some cases assembly language must be used (e.g., for issuing the mark time with the AST, for the AST routines themselves). The RSX-11M system services expected to be used by the AODS executive are as follows:

- SUSPEND/RESUME--initiates tasks to be executed
- ALTER PRIORITY--controls order of execution
- MARK TIME WITH AST--allows time slicing of tasks
- ABORT--allows removal of unwanted tasks from consideration
- CANCEL MARK TIME

4.1.1.4 Output Message Preparation

Six types of messages will be output from AODS:

- Residual Report
- DC Summary and Statistics Report
- Activities log

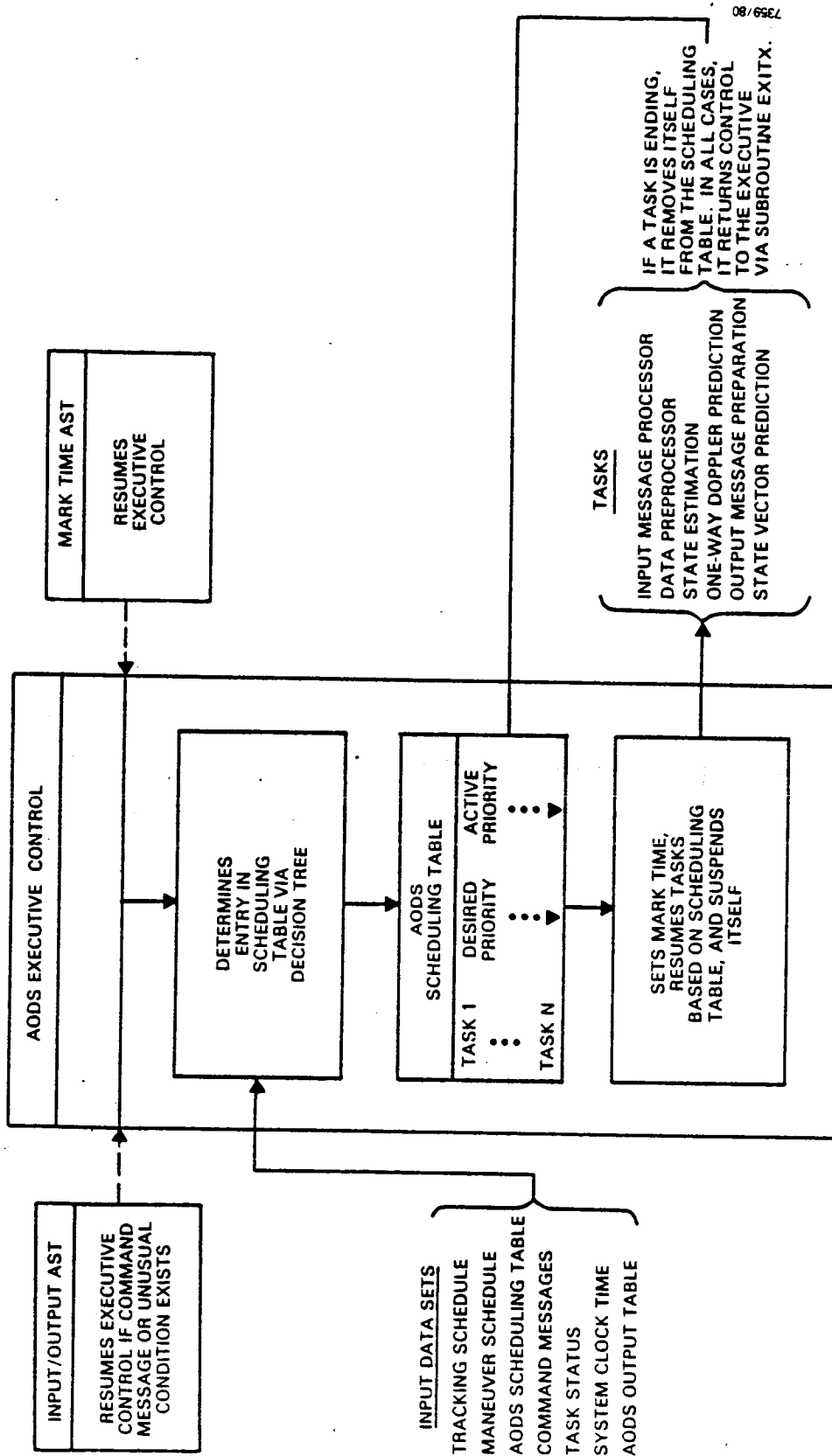


Figure 4-3. Executive Control

- Error messages
- One-way predicted Doppler observations
- Target state vectors

In the first-build AODS, the mechanisms for generating these messages will be implemented, but the messages themselves will be fixed in a global area. This implies that the stub tasks that perform output (e.g., state estimation task) will contain the proper communication and logic to inform the AODS executive to resume the output task. Figure 4-4 shows the output preparation procedure. To perform output, a task will mark the request in an output table and mark the message type locked. Upon examining the output table, the AODS executive will resume the output preparation task. The output preparation task will then

- Examine the output table for entries
- Move the data to the output message buffer; during the move, format the data for output
- When the move is complete, unlock the message
- Write the message using system routine QIO (see Section 4.1.1.5)
- Return to the first stop, looking for entries; if none exist, call EXITX¹

While the message type is locked, no data can overwrite it; therefore, tasks that must write into this buffer cannot progress until the unlock flag is set. The following tasks will require output service:

- State estimation task (Residual Report, DC Summary and Statistics Report)

¹EXITX will be a special routine through which the subordinate tasks will exit and inform the executive of the exit condition.

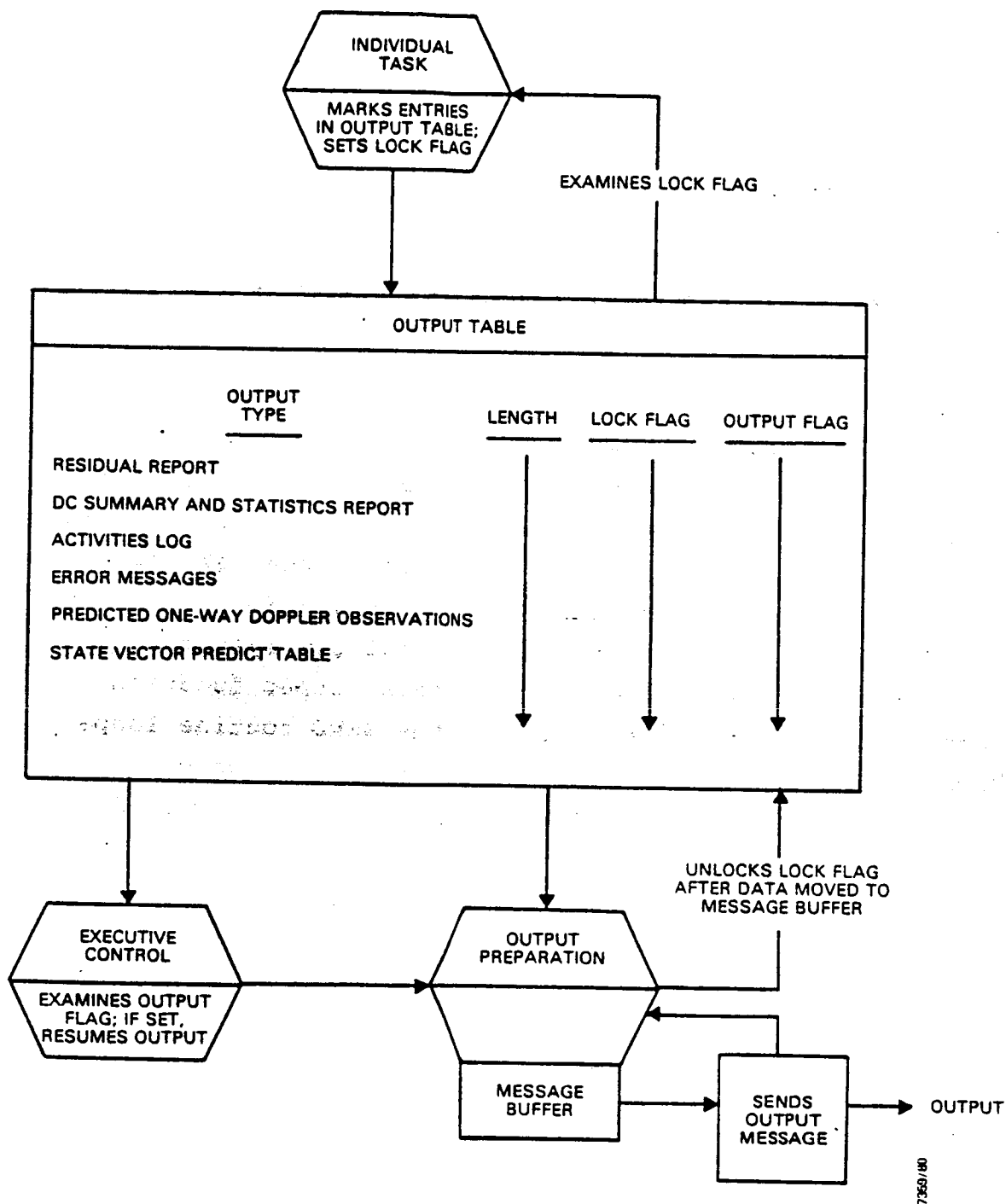


Figure 4-4. Output Preparation

- State vector prediction task (state vector predict table)
- One-way Doppler prediction task (predicted one-way Doppler observations)
- AODS executive control task (activities log, error messages)

The global COMMON area associated with the output preparation task is anticipated to be approximately 36,000 bytes, leaving 28,000 bytes for the output preparation code and internal buffer areas.

4.1.1.5 Output Message Transmission

The output message transmission component takes 256-byte messages from the message buffer (see Section 4.1.1.4) and performs the output operation. As with input data capture, this will be done in the first build as a SEND to the ADEPT downlink task. In future builds normal QIO system routines with ASTs will be used to perform this output function. Upon emptying the message buffer, the SEND routine loops back to prepare more output for processing. The output message preparation and transmission components will be implemented as one task image.

4.1.2 ADEPT IMPLEMENTATION (FIRST BUILD)

The environment simulator, ADEPT, will support the testing and evaluation of AODS. The first-build ADEPT will fully perform this function. In addition, interfaces for growth features such as data error simulation (e.g., noising data, message corruption, transmission failure) and truth model comparison will be established for future builds.

During this tasking period, the following work will be completed toward the eventual full implementation of the described first build:

- The detailed design of all components will be completed.
- The File Merge program will be implemented.
- The data preparation task (ADPREP) will be implemented as described in this section.
- A skeleton executive for the ground facility simulation task (ADSIM) will be implemented.

This level of implementation will be adequate to demonstrate the user interface required to set up a simulation and to exercise the major functions of the first-build AODS (primarily uplink and downlink communication).

ADEPT will support three major functions: data preparation, ground facility simulation, and output report generation and analysis. Figure 4-5 provides an overview of the proposed ADEPT data flow. A brief description of each ADEPT component is provided below:

- The File Merge program will merge tracking data tapes to prepare raw data files (disks) for simulation.
- The data preparation task (ADPREP) will perform the following functions:
 - Provide for the preparation of the following data sets: initialization table, TDRS vectors, estimation control parameters, maneuver schedules, tracking schedules, miscellaneous constants, station parameters, geopotential tables, atmospheric density tables, and timing coefficients--Each file will contain multiple

data blocks. Modification will be via interactive editing of displays showing default values. Any existing data block may be selected as the default. Data will be stored on disk in binary format.

- Provide for the initial creation of the simulation schedule from the observation selection parameters (e.g., interval between data passes, length of data passes, type(s) of data desired, data uplink frequency, burst size)-- In addition, the user will assign an uplink time to each data block which is to be uplinked during simulation. The constructed schedule will be stored on disk in card image format. This may be changed during the editing step (see below).
 - Provide verification of the simulation schedule--The simulation schedule will be printed with all selected data blocks. Selected blocks that are unavailable will be noted. Raw data will be checked for availability at each specified pass.
 - Provide for the modification of existing or newly created simulation schedules via a text editor.
- The ground facility simulation task (ADSIM) will perform the following functions:
 - Prepare and uplink data to AODS as specified by the simulation schedule--Data blocks will be selected by block number and type and uplinked at the specified times. Raw tracking data will be selected by type and time tag to

build each desired pass. Data error simulation (e.g., data noise, data bias, message corruption, transmission failure) will not be available in this build, but an appropriate "slot" will be provided for it in the logic.

- Report AODS and ADSIM status to the operator-- Several levels of status reporting will be available, ranging from display of all activity log and ADSIM debug messages to the display of only critical error messages. This output may optionally be diverted to the printer.
- Accept operator commands and uplinks to AODS-- AODS commands may also be included in the simulation schedule.
- Accept and act on "immediate" operator commands to the simulator--These are primarily immediate simulation schedule overrides. Downstream modification of the simulation schedule will be performed via concurrent operation of ADPREP. (Commands to ADSIM entered in this interactive fashion will not be written to the simulation schedule, but the action taken will be noted on the Simulation History file.) Other types of ADSIM commands include changes in reporting level, output to printer/terminal, and fast time on/off. Specifically, the following subset of the input data prepared by ADPREP will be interactively accessible to the user at the start of the simulator execution: start/stop/other commands, estimation control, output control,

message corruption/transmission failure probabilities, fast time options, and raw tracking data noise and bias.

- Receive and store output reports and predicted data downlinked by AODS--The DC Summary and Statistics Report will also be displayed/printed as received under the higher status reporting levels.
 - Write the Simulation History File--This is a complete real-time log of the simulation, including all data uplinked and received. (It may be on tape.)
 - In build 1, provide an experimental fast time capability to test the feasibility of packing idle time out of schedules via a handsnaking procedure with AODS
- The output report generation and analysis task (ADOUT) will perform the following functions:
 - Provide for the printing of operator-selected reports and predicted data--Selection of the desired report will be by type and time tag (available from higher levels of status reporting or the Simulation History file).
 - Provide a printout of the Simulation History file, which will be selected by time interval

As previously noted, certain interfaces will be designed into the system but not implemented into the first build. These include

- ADSIM
 - Data noise/bias
 - Code uplink to AODS
 - Data corruption/transmission failure

- ADOUT

- Comparison of predicted data with externally generated truth model, including graphics
- Full simulation history analysis, possibly including the ability to recreate runs

ADEPT will be built as follows:

- The File Merge program, ADPREP, ADSIM, and ADOUT will be built as separate tasks, with communication to the rest of the system achieved through shared data files.
- ADSIM will be built around already established AODS data capture programs (see Section 4.1.1.1).

4.2 FUTURE BUILDS

As noted earlier, the first-build AODS could evolve into a flight-qualified system. The steps for this evolution are expected to be as follows:

1. Add computational tasks (data preprocessing, state estimation, data base management, and one-way Doppler prediction) based on data obtained during the design activity and analysis performed outside of the Microprocessor Software Support task--These tasks should be able to be treated as strict applications programs¹ since the executive control performs most of the RSX-11M specific system services.

2. Move AODS to, and test it on, the LSI-11/23 micro-computer--The code developed and tested on the PDP-11/70 will be downloaded to the LSI-11/23. The code will be downloaded using the same asynchronous serial line over which AODS AND ADEPT communicate. First, the operating system

¹This means that the code developed can be 100 percent FORTRAN and unit tested outside the system. The algorithms themselves may be specially tailored to microcomputer limitations.

will be sent using a bootstrap program specially designed for the LSI-11/23. Next, the modified RSX-11S online task loader system routing will be used to load the individual AODS task images and data. Only two modifications to the AODS downloaded code are expected: (1) modification to data capture and send output routines to provide proper communication and (2) establishment of specific memory maps and resolution in some cases of specific locations.

3. Flight qualify the system--The final flight-qualified system will be much more automated than the LSI-11/23 test system. It will allow for moving the operating system and the AODS task images into a nonviolable memory. Upon startup of the LSI-11/23, the images will be bootstrapped into the microcomputer. The flight-qualified system will also have code modified based on the actual requirements of the mission. Changes are expected to be made to formats of input/output messages, communication protocols, and specific uplink commands.

APPENDIX A - STRUCTURED ANALYSIS

The following definitions are applicable to the data flow diagrams of the structured analysis techniques provided in Section 2:

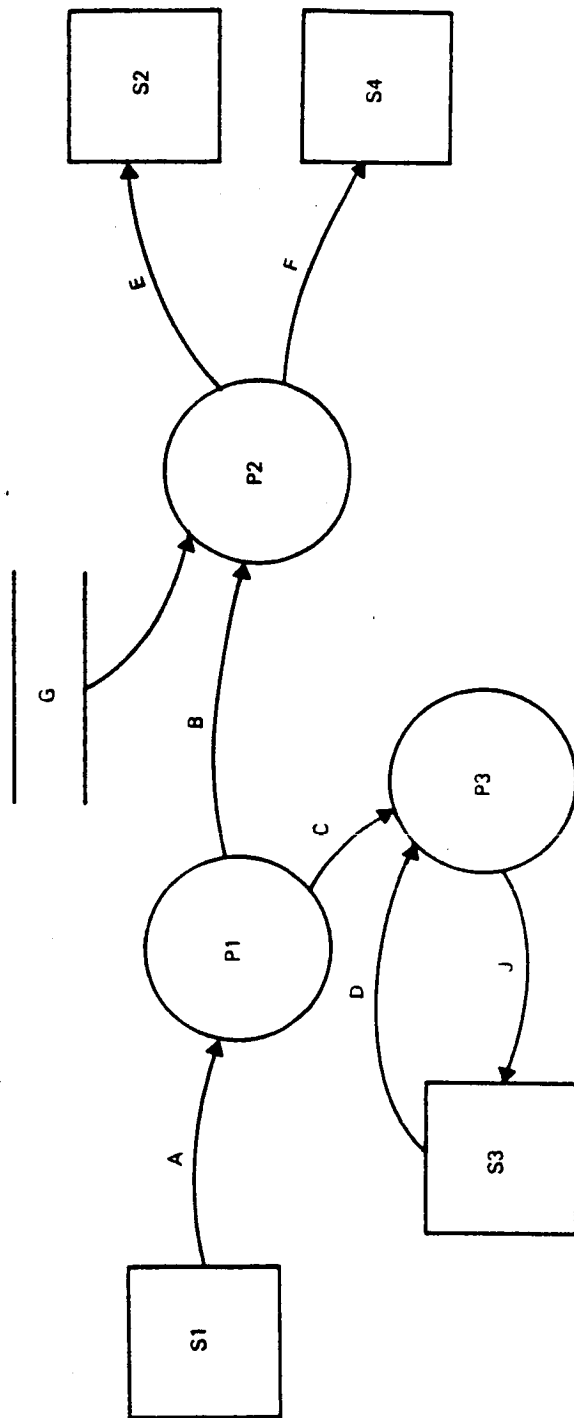
- Data flow is a pipeline through which packets of known composition flow.
- A process is a transformation of incoming data flow(s) into outgoing data flow(s).
- A data store is a time-delayed repository of information.
- A source or a sink is a system, person, or organization lying outside the context of study that is a net originator or receiver of data flows that are part of the study.

The ground rules for the data flow diagrams are as follows:

- Every data flow and data store must be named.
- Every process must be named with an active phrase describing the data transformation.
- Data flows exiting from a process must be derivable from data flows entering the process.
- Details of each process shall be described by another data flow diagram.
- Input and output shall be balanced between higher and lower level diagrams.

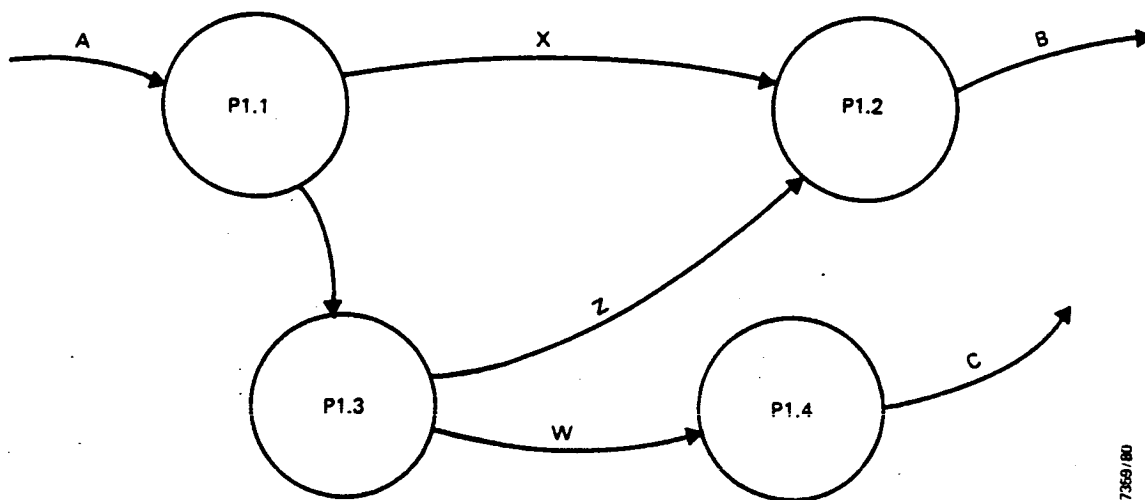
Figure A-1 is a top-level description of a system that has three processes (P1, P2, and P3), four sinks and/or sources (S1, S2, S3 and S4), one data store (G), and six data elements (A, B, C, D, E, and F).

Figure A-2 is the lower level description of process P1. The figure shows that at the next level process P1 may be



7359:80

Figure A-1. Top-Level Description of a System



7359/80

Figure A-2. Lower Level Description of a System

decomposed into subprocesses P1.1, P1.2, P1.3, and P1.4. Data elements A, B, and C must be shown at this level since they are the input and output of P1. Data elements X, Y and Z are data flows between the subprocesses. The processes at each level of the system can be completely described by a lower level diagram until the level of detail required for complete specification is reached.



APPENDIX B - DATA DICTIONARY

The data dictionary is necessary to make the data flow diagrams rigorous by defining specifically the content of each data element. To eliminate any ambiguity, the structured analysis technique utilizes a specification language called Structured English. This language has a limited vocabulary and syntax. For this study, however, conversational English has been used for ease of understanding by the reader.

The data dictionary provided on the following pages is organized according to major functions. The data elements are contained in the zero-, first-, and second-level diagrams.

B.1 DATA DICTIONARY FOR PROCESS 1: CONTROL SYSTEM

ACTIVITY LOG	=	Step-by-step account of all actions taken by AODS
CONTROL COMMAND	=	Special command from ground control that requires immediate action by control (commands are listed under Process 2 (Section B.2))
DIRECTIVES	=	Internally generated commands that direct the other processes (see Table B-1)
ERROR MESSAGE	=	PRIORITY MESSAGE stating that a severe error from which the system cannot recover has occurred
EXECUTION REQUEST	=	Next action to be taken based on execution status
GROUND CONTROL REQUEST	=	Next action to be taken as a result of a CONTROL COMMAND
MANEUVER SCHEDULE	=	Schedule of maneuvers of the active TDRSs and user spacecraft (see Process 2 (Section B.2))
PRIORITY MESSAGE	=	Message that must be sent immediately to ground control (e.g., error, idle time)
PREDICTED TDRS VECTOR AFTER MANEUVER	=	TDRS vector after the current maneuver (taken from the MANEUVER SCHEDULE)
PREDICTED USER SPACECRAFT VECTOR AFTER MANEUVER	=	User spacecraft state vector after the current maneuver (taken from the MANEUVER SCHEDULE)

Table B-1. Directives

DIRECTIVE NUMBER	DESTINATION PROCESS	DIRECTIVE
1	2	ACCEPT START COMMAND ONLY
2	2	ACCEPT ALL COMPLETE INPUT MESSAGES
3	3	BEGIN PREPROCESSING DATA
4	3	UPDATE TDRS ORBIT FILE USING NEW TDRS VECTOR
5	5	BEGIN ESTIMATION PROCESS
6	6	PREDICT ONE-WAY DOPPLER OBSERVATIONS
7	7	OUTPUT RESIDUAL REPORT
8	7	OUTPUT DC SUMMARY AND STATISTICS REPORT
9	7	OUTPUT STATUS (ACTIVITY LOG)
10	7	OUTPUT PREDICTED ONE-WAY DOPPLER OBSERVATIONS
11	7	OUTPUT PREDICTED TARGET SPACECRAFT STATE VECTORS
12	7	OUTPUT PRIORITY MESSAGE
13	7	GENERATE STATE VECTOR PREDICT TABLE
14	3	UPDATE TDRS INTEGRATION START DATA WITH NEW PRE- DICTED TDRS STATE AFTER THIS MANEUVER
15	4	PURGE DATA FILES
16	5	SUSPEND ESTIMATION
17	5	RESUME ESTIMATION
18	2	BEGIN PROCESSING INPUT MESSAGES
19	5	PERFORM HOUSEKEEPING IN ESTIMATOR (NECESSARY DUE TO USER SPACECRAFT MANEUVER)

7359/80

STATUS MESSAGES	=	System event message, system error message, or information message (see Table B-2)
SYSTEM IDLE MESSAGE	=	PRIORITY MESSAGE stating that the system has an excessive amount of idle time
TIME-SCHEDULED REQUEST	=	Next action to be taken according to the schedules and the clock
TRACKING INTERVAL	=	Next tracking interval over which one-way Doppler observations will be predicted
TRACKING SCHEDULE	=	Tracking schedule for predicting one-way Doppler observations (see Process 2)

Table B-2. Status Messages

MESSAGE NUMBER	GENERATIVE PROCESS	MESSAGE TYPE ^a	MESSAGE
1	2	ER	INPUT MESSAGE LOST
2	2	ER	INCOMPLETE INPUT MESSAGE
3	2	EV	OBSERVATION QUEUES FULL
4	2	EV	NEW TDRS VECTOR RECEIVED
5	2	EV	INITIALIZATION TABLE RECEIVED
6	2	EV	MANEUVER SCHEDULE RECEIVED
7	2	EV	ESTIMATION CONTROL PARAMETERS RECEIVED
8	2	EV	TRACKING SCHEDULE RECEIVED
9	2	EV	COMMAND RECEIVED
10	2	IN	NUMBER OF OBSERVATIONS RECEIVED
11	3	EV	TDRS ORBIT FILE UPDATED
12	3	EV	DATA PREPROCESSING COMPLETED
13	3	IN	NUMBER OF OBSERVATIONS WRITTEN TO OBSERVATIONS FILE
14	3	ER	ERROR IN TDRS PROPAGATION
15	3	ER	NO DATA FOR TDRS ID
16	5	ER	ERROR IN TARGET SPACECRAFT PROPAGATION
17	5	ER	TOO MANY OBSERVATIONS EDITED OUT
18	5	ER	NORMAL MATRIX CANNOT BE INVERTED
19	5	IN	NUMBER OF OBSERVATIONS EDITED
20	5	EV	ESTIMATION COMPLETED; NEW SOLUTION ATTAINED
21	5	EV	ESTIMATION COMPLETED; MAXIMUM ITERATIONS REACHED
22	6	ER	ERROR IN TDRS PROPAGATION
23	6	ER	ERROR IN TARGET SPACECRAFT PROPAGATION
24	6	EV	PREDICTION OF ONE-WAY DOPPLER OBSERVATIONS COMPLETED
25	7	EV	RESIDUAL REPORT OUTPUT
26	7	EV	DC STATISTICS AND SUMMARY REPORT OUTPUT
27	7	EV	STATUS (ACTIVITY LOG) OUTPUT
28	7	EV	STATE VECTORS OUTPUT
29	7	EV	PREDICTED ONE-WAY DOPPLER OBSERVATIONS OUTPUT
30	7	EV	PRIORITY MESSAGE OUTPUT
31	7	EV	STATE VECTOR PROPAGATED
32	7	ER	ERROR IN TARGET SPACECRAFT PROPAGATION
33	2	EV	INPUT CONTAINS DATA
34	2	EV	INPUT QUEUE FULL
35	2	EV	OBSERVATION QUEUES CONTAIN FULL DATA PASS
36	2	EV	MISCELLANEOUS CONSTANTS RECEIVED
37	2	EV	STATION CONSTANTS RECEIVED
38	2	EV	GEOPOTENTIAL TABLES RECEIVED
39	2	EV	ATMOSPHERIC DENSITY TABLE RECEIVED
40	2	EV	TIMING COEFFICIENTS RECEIVED
41	3	EV	TDRS INTEGRATION START DATA UPDATED
42	4	EV	DATA FILES PURGED
43	5	EV	HOUSEKEEPING COMPLETED
44	5	ER	NOT ENOUGH DATA FOR ESTIMATION

^aEV = SYSTEM EVENT; ER = ERROR; IN = INFORMATIVE.

7359/80

B.2 DATA DICTIONARY FOR PROCESS 2: PROCESS INPUT MESSAGES

ATMOSPHERIC DENSITY TABLES	=	Maximum and minimum density associated with altitude (100 to 1000 kilometers)
CODE	=	Partition of executable code + partition name
CONSTANTS	=	MISCELLANEOUS CONSTANTS
	+	STATION CONSTANTS
	+	GEOPOTENTIAL TABLES
	+	TIMING COEFFICIENTS
	+	ATMOSPHERIC DENSITY TABLE
CONTROL COMMANDS	=	Status request
	+	Start
	+	Stop
	+	Abort
	+	Suspend
	+	Continue
	+	Reboot
	+	Set clock
DATA	=	RAW OBSERVATION MESSAGES
	+	NEW TDRS VECTOR
	+	MANEUVER SCHEDULE
	+	TRACKING SCHEDULE
	+	INITIALIZATION TABLE
	+	CONSTANTS
	+	ESTIMATION CONTROL PARAMETERS
DIRECTIVES	=	Defined in Process 1 (Table B-1)

ESTIMATION CONTROL
PARAMETERS

- = Maximum iterations for estimation process
- + Observation weights
- + Sigma multiplier for editing
- + Convergence criteria
- + Maximum divergent iterations
- + Size of estimation time span
- + Observation weights
- + Residuals Report control switch
- + DC Summary and Statistics Report control switch

GEOPOTENTIAL TABLES

- = Order of harmonic field for TDRS and target
- + Degree of harmonic field for both
- + Point mass
- + Zonal harmonic coefficients (J₁ - J₁₅)
- + C harmonic coefficients (15-by-15)
- + S harmonic coefficients (15-by-15)

INITIALIZATION TABLE

- = A priori state parameters
- + A priori standard deviations
- + Reference time
- + Solve-for/consider map

INPUT MESSAGES

- = CONTROL COMMANDS + DATA + CODE

INPUT OBSERVATION
QUEUES

- = Usable bytes (see Appendix C) of the raw observation message, which are queued by type (1 = one-way TDRS; 2 = two-way TDRS; 3 = three-way TDRS; 4 = one-way SRE; 5 = two-way SRE)

MANEUVER SCHEDULE	=	(Maneuver time + spacecraft ID + predicted position and velocity) x 16
MISCELLANEOUS CONSTANTS	=	Drag switch for target
	+	Solar radiation pressure switch for TDRS
	+	Sun and Moon switches for TDRS and target
	+	Integration stepsizes for both
	+	Area of both spacecraft
	+	Mass of both spacecraft
	+	Solar flux constant
	+	Solar radiation pressure constant
	+	Rotation rate of Earth
	+	Equatorial radius of Earth
	+	Flattening coefficient of Earth
	+	Speed of light
	+	Reference time of Julian date
	+	Radians-to-degrees conversion constant
	+	Pilot frequency
	+	Refraction switch
	+	Timing biases for TDRSS and target
	+	Time pad for predicting one-way Doppler
	+	State vector predict table time interval
	+	State vector output frequency in predict table
RAM	=	Random-access memory
RAW OBSERVATION MESSAGE	=	Raw observation in the universal tracking data format (Appendix C)

STATION CONSTANTS	=	Total number of stations
	+	Station IDs
	+	Station positions
	+	Range bias
	+	Frequency bias
	+	Transponder delay
STATUS MESSAGES	=	Defined in Process 1 (Table B-2)
TIMING COEFFICIENTS	=	Coefficients of polynomials approximating USNO time difference data covering at least 1 year
TRACKING SCHEDULE (FOR ONE-WAY DOPPLER)	=	(Start and end times of each tracking interval + TDRS ID + output frequency) x 16

B.3 DATA DICTIONARY FOR PROCESS 3: PREPROCESS DATA

ACCEPTABLE OBSERVATION	=	Input observation that has the proper tracking configuration (i.e., right type, TDRS, station, ascending time order)
CONSTANTS	=	Defined in Process 2 (Section B.2)
DIRECTIVES	=	Defined in Process 1 (Table B-1)
INPUT OBSERVATION QUEUES	=	Defined in Process 2 (Section B.2)
NEW TDRS VECTORS	=	Defined in Process 2 (Section B.2)
OBSERVATION	=	Time tag
	+	Range measurement
	+	Doppler measurement
	+	Edit flags
	+	Ground antenna IDs
	+	Forward link ID and TDRS/ground terminal (GT) carrier frequency ID
	+	Return link ID and TDRS/GT carrier frequency ID
	+	TDRS frequency
	+	Access method
OBSERVATION DATA TIME SPAN	=	Start and end times of data in observation queues
PREDICTED TDRS VECTOR AFTER MANEUVER	=	Defined in Process 1 (Section B.1)
STATUS MESSAGES	=	Defined in Process 1 (Table B-2)

TDRS INTEGRATOR
START DATA

= TDRS position and velocity at
final propagation time
+ Final propagation time (T)
+ TDRS ID
+ Velocities and accelerations at
time $T-\Delta T$, $T-2\Delta T$, $T-3\Delta T$,
 $T-4\Delta T$, $T-5\Delta T$, $T-6\Delta T$, $T-7\Delta T$,
 $T-8\Delta T$
+ Save counter

TDRS VECTOR

= TDRS position and reference
time + TDRS ID

B.4 DATA DICTIONARY FOR PROCESS 4: MANAGE DATA BASE

DIRECTIVE	=	Defined in Process 1 (Table B-1)
LAST OBSERVATION TIME	=	Time tag of the last observation in the file
OBSERVATION	=	Defined in Process 3 (Section B.3)
OBSERVATIONS FILE	=	File containing up to 500 observation pairs
OBSERVATION READ ADDRESS	=	Location in memory of the observation record that is to be read
OBSERVATION WRITE ADDRESS	=	Location in memory where the observation record is to be written
STATUS MESSAGES	=	Defined in Process 1 (Table B-2)
TDRS ORBIT FILES	=	Files containing TDRS vectors at an even time spacing over each observation time span for each TDRS
TDRS READ ADDRESS	=	Location in memory of the TDRS record that is to be read
TDRS REQUEST TIME	=	Reference time of the TDRS vector(s) that are to be read
TDRS VECTOR(S)	=	TDRS vectors surrounding the TDRS request time that may be used for interpolation
TDRS WRITE ADDRESS	=	Location in memory where the TDRS vector is to be written

B.5 DATA DICTIONARY FOR PROCESS 5: ESTIMATE STATE

BATCH ESTIMATION MATRICES	=	Normal equation and matrix used to solve for the state correc- tion
COMPUTED OBSERVATION	=	OBSERVATION + computed range and Doppler based on the cur- rent state
CONSTANTS	=	Defined in Process 2 (Sec- tion B.2)
DC STATISTICS	=	Current weighted rms + Predicted weighted rms + Previous weighted rms + Smallest weighted rms + Relative change in rms + Penalty + Convergence message + Start time + End time
DC SUMMARY	=	Previous state + Current state + STATE CORRECTION + Standard deviations + A priori state
DC SUMMARY AND STATISTICS REPORT	=	DC STATISTICS + EDITING STATISTICS + DC SUMMARY
DIRECTIVE	=	Defined in Process 1 (Table B-1)
EDITING STATISTICS	=	Number of observations available + number of observations used + number of observations edited by n sigma editing
ESTIMATION PARAM- ETERS	=	SOLVE-FOR PARAMETERS + ESTIMATION CONTROL PARAMETERS (see Process 2 (Section B.2))

NEW STATE SOLUTION	=	UPDATED STATE VECTOR when the estimation has converged
LAST OBSERVATION TIME	=	Defined in Process 4 (Section B.4)
OBSERVATION	=	Defined in Process 3 (Section B.3)
OBSERVATION EDIT TAGS	=	Indicator associated with each computed observation that shows whether the residual is within a specified range
OBSERVATION PARTIALS	=	Partial derivatives of the observations (range and Doppler) with respect to the solve-for state at the observation time
OBSERVATION RESIDUAL	=	Difference between the observed and computed measurements (O-C)
OBSERVATION TIME	=	Time tag on observation
SOLVE-FOR PARAMETERS	=	Solve-for state + standard deviations
	+	Observation weights
	+	Solve-for/consider map
	+	A priori state
STATE CORRECTION VECTOR	=	Estimated correction for each state parameter at epoch
STATE TRANSITION MATRIX	=	Matrix used to propagate the observation partials from the observation time to epoch
STATUS MESSAGES	=	Defined in Process 1 (Table B-2)
TDRS REQUEST TIME	=	Defined in Process 4 (Section B.4)
TARGET STATE VECTOR	=	Target spacecraft position and velocity at the observation time

TDRS VECTOR(S) = Defined in Process 3 (Section B.3)

UPDATED STATE VECTOR = (Solve-for state + state correction)

B.6 DATA DICTIONARY FOR PROCESS 6: PREDICT ONE-WAY DOPPLER OBSERVATIONS

CONSTANTS	=	Defined in Process 2 (Section B.2)
DIRECTIVE	=	Defined in Process 1 (Table B-1)
NEXT OBSERVATION TIME	=	Time at which next observation will be predicted
ONE-WAY PREDICTED DOPPLER OBSERVATIONS	=	Doppler measurement + station ID + TDRS ID + time tag
NEW STATE SOLUTION	=	Defined in Process 5 (Section B.5)
PROPAGATED TDRS VECTOR	=	TDRS position and velocity propagated to observation time
STATUS MESSAGE #	=	Defined in Process 1 (Table B-2)
TARGET STATE VECTOR	=	Position and velocity of target spacecraft propagated to next observation time
TDRS VECTOR	=	Defined in Process 3 (Section B.3)
TRACKING INTERVAL	=	Defined in Process 1 (Section B.1)

B.7 DATA DICTIONARY FOR PROCESS 7: PERFORM OUTPUT

ACTIVITY LOG	=	Defined in Process 1 (Section B.1)
CONSTANTS	=	Defined in Process 2 (Section B.2)
DC RESIDUALS REPORT	=	Defined in Process 5 (Section B.5)
DC SUMMARY AND STATISTICS REPORT	=	Defined in Process 5 (Section B.5)
DIRECTIVES	=	Defined in Process 1 (Table B-1)
ENCODED DATA MESSAGE	=	Data encoded in output message format
ENCODED PRIORITY MESSAGE	=	PRIORITY MESSAGE encoded in output message format
ENCODED REPORT MESSAGE	=	One part of the report encoded in output message format
NEW STATE SOLUTION	=	Defined in Process 5 (Section B.5)
ONE-WAY PREDICTED DOPPLER OBSERVATIONS	=	Defined in Process 6 (Section B.6)
OUTPUT MESSAGES	=	All messages output from AODS in a form that can be put on the transmission line
OUTPUT TARGET STATE VECTORS	=	Table of position and velocity vectors of the target spacecraft that are time tagged; the vectors are generated at a specified frequency over a specified time interval
RESIDUAL REPORT	=	Defined in Process 5 (Section B.5)
STATUS MESSAGES	=	Defined in Process 1 (Table B-2)

APPENDIX C - INPUT AND OUTPUT MESSAGE FORMATS

This appendix contains the input and output message formats through which AODS will communicate with the external world. Figure C-1 shows the standard data transmission format that will be used for both input and output. A definition of terms used in Figure C-1 and throughout the appendix is provided below:

<u>Term</u>	<u>Definition</u>
Transmission	Set of one or more blocks of data that are transmitted contiguously. A transmission is always terminated by an end-of-transmission record (all 1s)
Block	Set of one or more data records that contain the same type of data
Record	A 256-byte record containing a header (20 bytes) and one or more frames of data (see Figure C-2)
Frame	One entity of data
Header	A 20-byte header frame that describes the contents of the record

C.1 INPUT (UPLINK) MESSAGE FORMATS

This section contains the input message formats through which data and commands will be uplinked to AODS. Section C.1.1 specifies the format of the record header (first 20 bytes), which is common to all input records. Section C.1.2 through C.1.13 specify the message block attributes and the frame format for each type of input data and command.

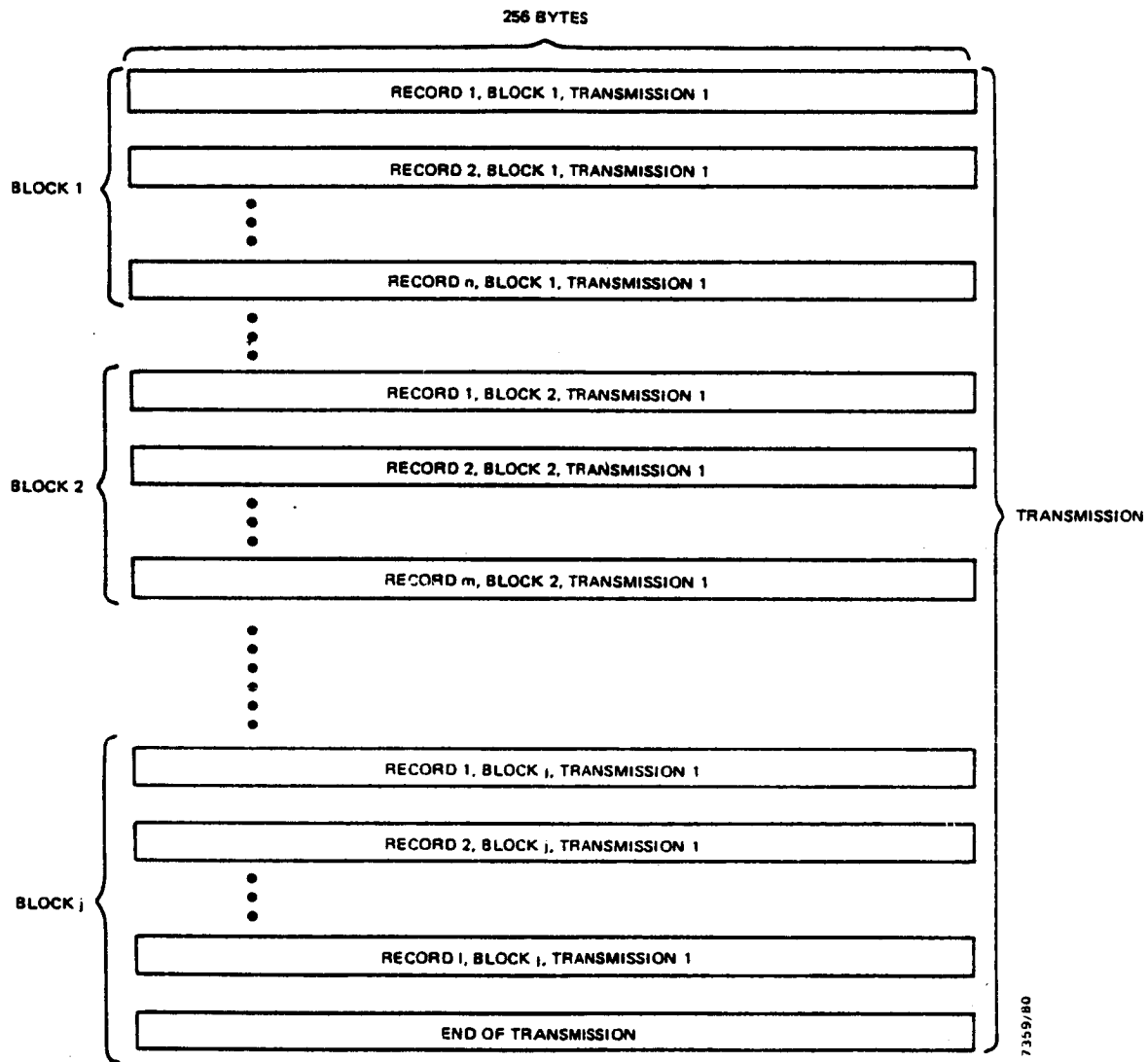


Figure C-1. Data Transmission Format

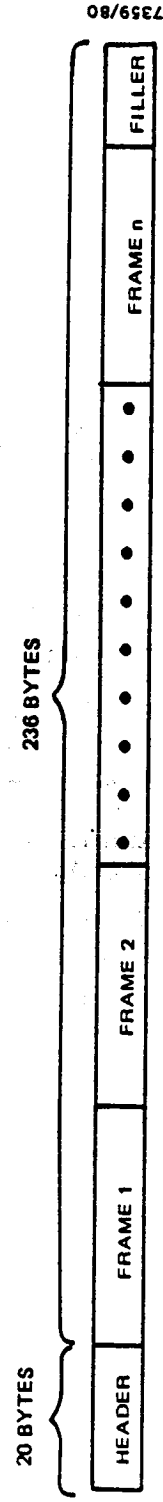


Figure C-2. Transmission Record Format

C.1.1 HEADER

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
IDSC	Byte	1	Spacecraft ID
IDEX	Byte	1	Experiment ID
INTYPE	Byte	1	Type of input: = 1, data = 2, code = 3, commands
INDATA	Byte	1	Type of data: = 1, raw observations = 2, initialization table = 3, new TDRS vector = 4, estimation control parameters = 5, maneuver schedule = 6, tracking schedule = 7, miscellaneous constants = 8, station constants = 9, geopotential tables = 10, atmospheric drag = 11, timing coefficients
NBLOCK	Byte	1	Running number of record in block
MBLOCK	Byte	1	Total number of records in block
NTRAN	I*2	1	Running number of record in transmission
MTRAN	I*2	1	Total number of records in transmission
NSIZE	I*2	1	Number of bytes used in record
TTRAN	R*8	1	Time of transmission (seconds from reference time)

C.1.2 RAW OBSERVATION DATA MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = 1 raw observation (75 bytes)

1 record = header + 3 frames + fill

256 = 20 + 3 x 75 + fill

256 = 245 + fill

1 block = 1 or more records (defined during transmission)

FRAME FORMAT^a:

<u>Data Field</u>	<u>Format^b</u>	<u>Contents of Field</u>	<u>Reduced Observation</u>	
			<u>Type</u>	<u>Bytes</u>
Bytes 1-5	H	Fill	-	0
Byte 6	B	Last two digits of current year	Byte	1
Bytes 7-8	B	Support Identification Code (SIC) of user or TDRS providing return link	-	0
Bytes 9-10	B	Vehicle ID (VID) of target	:-	0
Bytes 11-14	B	Time tag (seconds of year)	Real	4
Bytes 15-18	B	Time tag (microseconds of second)	Real	4
Bytes 19-22	B	Return link azimuth angle	-	0
Bytes 23-26	B	Return link elevation angle	-	0
Bytes 27-32	B	Range measurement nanoseconds)	Real	8
Bytes 33-38	B	Doppler count	Real	8
Bytes 39-40	H	Fill (0016)	-	0

^aThis is the universal tracking data format, taken from Reference 9.

^bH = hexadecimal, B = binary, D = discrete.

<u>Data Field</u>	<u>Format</u>	<u>Contents of Field</u>	<u>Reduced Observation</u>	
			<u>Type</u>	<u>Bytes</u>
Bytes 41-44	B	Reference frequency used in Doppler extraction	Integer	4
Byte 45	H	Fill (60 ₁₆)	-	0
Byte 46	B	TDRS ground antenna ID associated with TDRS providing forward link	Byte	1
Byte 47	H	Fill (60 ₁₆)	-	0
Byte 48	B	Return link ground antenna ID associated with TDRS	Byte	1
Byte 49		TDRS IDs:		
Bits 8-5	H	Forward link	Byte	1
Bits 4-1	H	Return link	Byte	1
Byte 50				
Bits 8-4	B	Multiple-access return link ID; binary ID of radio frequency (RF) Beam-forming equipment	-	0
Bit 3	B	TDRS tracking data only indicator	-	0
Bits 2-1	B	Tracking service configuration	-	0
Byte 51		Data validity field:		
Bit 1	D	Validity of range	Byte	1
Bit 2	D	Validity of Doppler	Byte	1
Bit 3	D	Validity of return link antenna angles	-	0

<u>Data Field</u>	<u>Format</u>	<u>Contents of Field</u>	<u>Reduced Observation</u>	
			<u>Type</u>	<u>Bytes</u>
Byte 52	H	Frequency band (most significant digit (MSD) and service type (least significant digit (LSD))	-	0
Byte 53				
MSD	H	Tracker type	Byte	1
Bit 4	D	End of track	Byte	1
Bit 3	D	0 indicates data in sample rate field is seconds between tracking samples	Byte	1
Byte 53, bit 2 through byte 54, bit 1	B	Seconds between tracking samples	R	4
Byte 55				
Bit 8	D	TDRS orientation validity	-	0
Bit 7	D	RF beam orienta- tion validity	-	0
Bits 6-4	B	Forward link ID and TDRS/GT carrier fre- quency ID	Byte	1
Bits 3-1	B	Return link ID and TDRS/GT carrier fre- quency ID	Byte	1
Byte 56				
Bits 6-5	D	User bit rate indicator	-	0
Bits 4-1	B	TDRS tracking data transponder ID	-	0
Bytes 57-58	B	TDRS yaw	-	0
Bytes 59-60	B	TDRS roll	-	0

<u>Data Field</u>	<u>Format</u>	<u>Contents of Field</u>	<u>Reduced Observation</u>	
			<u>Type</u>	<u>Bytes</u>
Bytes 61-62	B	TDRS pitch	-	0
Bytes 63-65	H	RF beam azimuth	-	0
Bytes 66-68	H	RF beam elevation	-	0
Bytes 69-75	H	Fill (00 ₁₆ , 04 ₁₆ , 0F ₁₆)	-	0

C.1.3 INITIALIZATION TABLE INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = 1 initialization table (188 bytes)
1 record = 1 header + frame + fill
 256 = 20 + 188 + fill
 256 = 208 + fill
1 block = 1 record

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
REFTM	R*8	1	Reference time
REFAPR	R*8	10	A priori state vector
REFSTD	R*8	10	A priori standard deviation
MAP	I*2	10	Solve-for consider map

C.1.4 NEW TDRS VECTOR(S) INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = new TDRS vector for 1 TDRS (60 bytes)
1 record = header + 1 frame + fill
256 = 20 + 60 + fill
256 = 80 + fill
1 block = 1 to 3 records (defined at transmission)

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
TDRTIM	R*8	1	TDRS reference time (YYMMDDHHMMSS.SS)
TDRSX	R*8	6	New TDRS position and velocity vectors
IDTDRS	I*2	1	TDRS ID
INTYPE	I*2	1	Type of input vector: = 0, new estimate of TDRS vector = 1, update to previous TDRS maneuver

C.1.5 ESTIMATION CONTROL PARAMETERS INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = estimation control parameters set (66 bytes)
 1 record = 1 frame + fill
 256 = 66 + fill
 1 block = 1 record

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
DCSPAN	R*4	1	Estimation time span (size of batch of data in seconds)
SPARE	I*2	1	Spare
SIGMA	R*4	1	Sigma multiplier for editing
CONVRG	R*8	1	Convergence criteria
MAXITR	I*2	1	Maximum number of iter- ations to be performed
MAXDIV	I*2	1	Maximum number of di- vergent iterations allowed
OBSSTD(I,J)	R*4	2,5	Observtion standard de- viations: I = measurement type (= 1, range; = 2, Doppler) J = observation type (= 1, one-way TDRSS; = 2, two-way TDRS; = 3, three-way TDRSS = 4, one-way SRE; = 5, two-way SRE)
IROUT	I*2	1	Residual Report output control switch: = 0, no report = 1, report generated after last itera- tion on a batch of data = 2, report generated after first and last iterations on a batch of data

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
IDCOUT	I*2	1	DC Summary and Statistics Report output control switch: = 0, no report = 1, report generated after last iteration on a batch of data = 2, report generated after every iteration

C.1.6 MANEUVER SCHEDULE INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = 1 scheduled maneuver (58 bytes)
1 record = header + 4 frames + fill
 256 = 20 + 232 + fill
 256 = 252 + fill
1 block = 4 records

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
TIM01	R*8	1	Time of maneuver (YYMMDDHHMMSS.SS)
XM01	R*8	6	Predicted state (posi- tion and velocity) after maneuver
MSID01	I*2	1	ID of maneuvered space- craft (TDRS ID for TDRS, SIC and VID for user spacecraft)

C.1.7 TRACKING SCHEDULE INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = schedule for 1 tracking interval (22 bytes)
1 record = header + 8 frames + fill
256 = 20 + 8 x 22 + fill
256 = 20 + 176 + fill
256 = 196 + fill
1 block = 2 records

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
STIME	R*8	1	Start time of tracking interval (YYMMDDHHMMSS.SS)
ETIME	R*8	1	End time of tracking interval (YYMMDDHHMMSS.SS)
TDRSID	I*2	1	ID of TDRS to be used for one-way Doppler prediction during this interval
OBSFRQ	R*4	1	Observation frequency

C.1.8 MISCELLANEOUS CONSTANTS INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = set of constants (236 bytes)
 1 record = header + 1 frame = 256
 1 block = 1 record

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
NDRAG	Byte	1	Drag switch ^a for target
NSOLRD	Byte	1	Solar radiation pressure switch ^a for TDRS
NSUN(I)	Byte	2	Sun switches: ^a I = 1, target I = 2, TDRS
NOOM(I)	Byte	2	Moon switches: ^a I = 1, target I = 2, TDRS
STEPSZ(I)	R*4	2	Integration stepsizes: I = 1, target I = 2, TDRS
SCAREA(I)	R*4	2	Area of spacecraft: I = 1, target; I = 2, TDRS
SCMASS(I)	R*4	2	Mass of spacecraft: I = 1, target I = 2, TDRS
SFLUX	R*4	1	Solar flux value
SOLRAD	R*4	1	Solar radiation pressure
REFJUL	R*8	1	Reference time of Julian date
OMEGA	R*8	1	Rotation rate of Earth
EQTRAD	R*8	1	Equatorial radius
FLAT	R*8	1	Flattening coefficient
PI	R*8	1	π

^aSwitches: 1 = on, 0 = off.

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
VLITE	R*8	1	Speed of light
RTD	R*8	1	Radians-to-degrees conversion constant
TFREQ(I)	R*8	5	Table used to compute pilot frequency for the following access methods: I = 1, multiple-access (MA) I = 2, S-band single-access link 1 (SSA1) I = 3, SSA2 I = 4, K-band single-access link 1 (KSA1) I = 5, KSA2
TBIASR(I)	R*8	3	Timing bias for TDRS: I = 1, TDRS I I = 2, TDRS II I = 3, TDRS III
TBIASS	R*8	1	Timing bias for user spacecraft
IFRAC	I*2	1	Refraction switch: = 0, off = 1, on
TPAD	R*4	1	Time pad for output of predicted one-way Doppler data (minutes)
SPINT	R*4	1	State vector predict table size (minutes) (default = 30 minutes)
SPFREQ	R*4	1	State vector frequency in predict table (minutes) (default = 1 minute)
CSPARES	R*4	14	Spare locations for constants

C.1.9 STATION PARAMETERS INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

Set of constants = 1002 bytes

Record 1 = header + frame 1 (234 bytes) = 254 bytes + fill
Record 2 = header + frame 2 (192 bytes) = 212 bytes + fill
Record 3 = header + frame 3 (192 bytes) = 212 bytes + fill
Record 4 = header + frame 4 (192 bytes) = 212 bytes + fill
Record 5 = header + frame 5 (192 bytes) = 212 bytes + fill

1 block = 5 records

FRAME 1 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
NSTA	I*2	1	Total number of stations used
IDSTA(J)	I*2	20	Station IDs in order corresponding to positions
STAT(I,J)	R*8	6,4	Constants for station J, where J = 1 through 4: I = 1, Earth-fixed position component-X I = 2, Earth-fixed position component-Y I = 3, Earth-fixed position component-Z I = 4, range bias I = 5, frequency bias I = 6, transponder delay

FRAME 2 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
STAT(I,J)	R*8	6,4	Constants for station J, where J = 5 through 8 (see frame 1 format, above)

FRAME 3 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
STAT(I,J)	R*8	6,4	Constants for station J, where J = 9 through 12 (see frame 1 format, above)

FRAME 4 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
STAT(I,J)	R*8	6,4	Constants for station J, where J = 13 through 16 (see frame 1 format, above)

FRAME 5 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
STAT(I,J)	R*8	6,4	Constants for station J, where J = 17 through 20 (see frame 1 format, above)

C.1.10 GEOPOTENTIAL TABLES INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

Total set of constants = 1004 bytes

Record 1 = header + frame 1 (232 bytes) = 256 bytes

Record 2 = header + frame 2 (200 bytes) = 220 bytes

Record 3 = header + frame 3 (200 bytes) = 220 bytes

Record 4 = header + frame 4 (200 bytes) = 220 bytes

Record 5 = header + frame 5 (200 bytes) = 220 bytes

1 block = 5 records

FRAME 1 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
MORD(I)	Byte	2	Order of harmonic field I = 1, target I = 2, TDRS
MDEG(I)	Byte	2	Degree of harmonic field I = 1, target I = 2, TDRS .
GM	R*8	1	Point mass
XJ	R*4	15	Zonal harmonics (J ₁ through J ₁₅)
CS	R*4	40	First 40 C- and S- harmonic coefficients (C-harmonic coeffi- cients in lower trian- gle of 15-by-16 matrix; S-harmonic coefficients in upper triangle of 15-by-16 matrix) for 15-by-15 geopotential model

FRAME 2 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
CS	R*4	50	Next 50 C- and S- harmonic coefficients (C-harmonic coeffi- cients in lower trian- gle; S-harmonic co- efficients in upper triangle) of 15-by-15 model

FRAME 3 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
CS	R*4	25	Next 50 C- and S-harmonic coefficients (C-harmonic coefficients in lower triangle; S-harmonic coefficients in upper triangle) of 15-by-15 model

FRAME 4 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
CS	R*4	50	Next 50 C- and S-harmonic coefficients (C-harmonic coefficients in lower triangle; S-harmonic coefficients in upper triangle) of 15-by-15 model

FRAME 5 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
CS	R*4	50	Last 50 C- and S-harmonic coefficients (C-harmonic coefficients in lower triangle; S-harmonic coefficients in upper triangle) of 15-by-15 model

C.1.11 ATMOSPHERIC DENSITY TABLES INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

TOTAL SET OF DATA = 662 bytes

Record 1 = header + frame 1 (234 bytes) = 254 + fill

Record 2 = header + frame 2 (224 bytes) = 244 + fill

Record 3 = header + frame 3 (144 bytes) = 164 + fill

1 BLOCK = 3 records

FRAME 1 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
NDENS	I*2	1	Number of entries in density table
NALT(J)	I*2	60	Altitude associated with density intervals (in ascending order)
DENSTY(I,J)	R*4	2,14	First 14 intervals in density table: I = 1, minimum density associated with NALT(J) I = 2, maximum density associated with NALT(J)

FRAME 2 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
DENSTY(I,J)	R*4	2,28	Next 28 intervals in density table

FRAME 3 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
DENSTY(I,J)	R*4	2,18	Last 18 intervals in density table

C.1.12 TIMING COEFFICIENTS INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = total set of data (226 bytes)
1 record = header + 1 frame + fill
256 = 20 + 226
256 = 246 + fill
1 block = 1 record

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
NDAYS	I*2	1	Number of polynomials used in TCOEFF
TCOEFF(I,J)	R*4	7,8	Coefficients of polynomials approximating USNO time difference data: I = 1, modified Julian date associated with polynomial J I = 2, coefficient a _{i1} in polynomial J I = 3, coefficient a _{i2} in polynomial J I = 4, coefficient a _{i3} in polynomial J I = 5, coefficient a _{i4} in polynomial J I = 6, coefficient a _{i5} in polynomial J I = 7, coefficient a _{i6} in polynomial J

C.1.13 CONTROL COMMAND INPUT MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = 1 command (202 bytes)

1 record = header + 1 frame = 20 + 202 + fill = 222 + fill

1 block = 1 record

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
ICTYPE	I*2	1	Type of command: = 1, reboot = 2, abort = 3, stop = 4, start = 5, suspend = 6, continue = 7, status request = 8, set clock
COMMAND(I)	Byte	200	Contents of command (depends on type of command)

C.2 OUTPUT (DOWNLINK) MESSAGE FORMATS

This section contains the output message formats through which data, reports, and informative messages will be downlinked from AODS. Section C.2.1 specifies the format of the record header, which is common to all output records. Sections C.2.2 through C.2.8 specify the message block attributes and frame formats for each type of output data, report, and message.

C.2.2 HEADER

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
IDSC	Byte	1	Spacecraft ID
IDEX	Byte	1	Experiment ID
OUTYPE	Byte	1	Type of output: = 1, spacecraft vectors = 2, Doppler observations = 3, error message = 4, activity log = 5, DC Summary and Statistics Report = 6, DC Residuals Report
	Byte	1	Blank
NBLOCK	I*2	2	Running number of records in a block
NTRAN	I*2	2	Running number of records in a transmission
NSIZE	I*2	:2	Record size in bytes
	I*2	2	Blank
TTRAN	R*8	8	Time of transmission (seconds from reference time)

C.2.3 OUTPUT USER SPACECRAFT STATE VECTORS

MESSAGE BLOCK ATTRIBUTES:

1 frame = 1 state vector (56 bytes)
1 record = header + 4 frames + fill
256 = 20 + 224 + fill
256 = 244 + fill
1 block = 1 or more records

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
TTAG	R*8	1	Time tag
PVEC	R*8	3	Position vector (x, y, z)
VVEC	R*8	3	Velocity vector (\dot{x} , \dot{y} , \dot{z})

C.2.4 OUTPUT ONE-WAY DOPPLER OBSERVATIONS

MESSAGE BLOCK ATTRIBUTES:

1 frame = 1 observation (22 bytes)
1 record = header + 10 frames + fill
256 = 20 + 220 + fill
256 = 240 + fill
1 block = 1 or more records

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
OBTYPE	Byte	1	Observation type: = 1, TDRSS one-way = 4, S-band one-way
IDTDRS	Byte	1	TDRS ID
IDSTAF	I*2	1	Forward link station ID
IDSTAR	I*2	1	Return link station ID
OBTIME	R*8	1	Time tag
DOPLR1	R*8	1	Doppler observation

C.2.5 OUTPUT ERROR MESSAGE

MESSAGE BLOCK ATTRIBUTES:

1 frame = 1 error message (46 bytes)
1 record = header + 1 frame + fill
 256 = 20 + 46 + fill
 256 = 66 + fill
1 block = 1 record

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
TERR	R*8	1	Time of error
ERRMSG	Byte	38	Message

C.2.6 OUTPUT FROM ACTIVITY LOG

1 frame = 1 message (50 bytes)
 1 record = header + 4 frames + fill
 256 = 20 + 200 + fill
 256 = 220 + fill
 1 block = 20 records

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
TMSG	R*8	1	Time message entered log
MSGTYP	Byte	1	Type of message: = 0, error = 1, event = 2, informative = 3, directive = 4, command
IATYPE	Byte	1	Source on which action taken was based: = 1, execution = 2, ground control request = 3, schedule and system clock
MSG	Byte	40	Message

C.2.7 DC SUMMARY AND STATISTICS REPORT

MESSAGE BLOCK ATTRIBUTES:

Whole report = 524 bytes

Record 1 = header + frame 1 + fill = 20 + 160 + fill = 256

Record 2 = header + frame 2 + fill = 20 + 160 + fill = 256

Record 3 = header + frame 3 + fill = 20 + 184 + fill = 256

1 block = record 1 + record 2 + record 3 = 3 records

FRAME 1 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
TEPOCH	R*8	1	Epoch
STIME	R*8	1	Start time of estimation data span
ETIME	R*8	1	End time of estimation data span
RMSCNR	R*8	1	Current weighted rms
RMSPRE	R*8	1	Predicted weighted rms
RMSLST	R*8	1	Previous weighted rms
RMSMIN	R*8	1	Smallest weighted rms
RMSCHG	R*8	1	Relative change in rms
PENLTY	R*8	1	Penalty
CONVRG	R*8	1	Convergence criteria
XPREV	R*8	10	Previous state vector

FRAME 2 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
XCURR	R*8	10	Current state vector
XAPR	R*8	10	A priori state vector

FRAME 3 FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
XSTD	R*8	10	Standard deviations for state vector
XUPD	R*8	10	State correction vector
ISTATE	I*2	10	Parameter numbers
NSTATE	I*2	1	Number of solve-for parameters
NTOTAL	I*2	1	Total number of observations available
NUSED	I*2	1	Number of observations used
NITER	I*2	1	Iteration number
NBATCH	I*2	1	Batch number

C.2.8 DC RESIDUALS REPORT

MESSAGE BLOCK ATTRIBUTES:

1 frame = 1 line of report (48 bytes)
 1 record = header + 4 frames + fill
 256 = 20 + 192 + fill
 256 = 212 + fill
 1 block = 125 records

FRAME FORMAT:

<u>Variable</u>	<u>Type</u>	<u>Dimension</u>	<u>Description</u>
IOBTYP	Byte	1	Observation type: = 1, TDRSS one-way = 2, TDRSS two-way = 3, TDRSS hybrid = 4, S-band one-way = 5, S-band two-way
IEDIT(I)	Byte	2	Edit flag (I = 1, range; I = 2, Doppler): = 0, not edited = 1, edited by DC = 2, edited during pre-processing
ITDRSF	Byte	1	TDRS ID (forward link)
ITDRSR	Byte	1	TDRS ID (return link)
ISTATF	Byte	1	Forward link station ID
ISTATR	Byte	1	Return link station ID
ISPARE	Byte	1	Spare location
OBTIME	R*8	1	Time tag
OBS(1)	R*8	1	Computed range observation
OBS(2)	R*8	1	Computed Doppler observation
RESID(1)	R*4	1	Range residual
RESID(2)	R*4	1	Doppler residual
RTS(1)	R*4	1	Ratio to sigma for range (range/standard deviation)
RTS(2)	R*4	1	Ratio to sigma for Doppler (Doppler/standard deviation)

APPENDIX D - PROCEDURES FOR MODIFYING AODS REQUIREMENTS

Since a formal methodology was used to define the AODS requirements, they are easily modified. Three parts of the requirements definition could change: (1) data, (2) processes, and (3) functional requirements. A change in one of these could, but would not necessarily, affect the other two. To ensure that all effects of a requirements modification are made apparent, the changes should be made as described below.

To change a data definition, the data item should be located in the data dictionary (Appendix B) and modified. Through the data dictionary, the processes that act upon that data item should be located and changed if necessary. The functional requirements affected can then be located through the Process/Requirements Map (Table 2-2) and changed where necessary.

To modify a process definition, the process should be located in the data flow diagrams to determine whether the definition of any of its associated data flows is impacted. If so, the appropriate change(s) should be made in the data dictionary. Also, the functional requirements that are affected should be located through the Process/Requirements Map (Table 2-2) and modified where necessary.

To modify a functional requirement, the processes affected should be located through the Requirement/Processes Map (Table 2-1) and modified where necessary. Any data flows that are affected can then be located and modified accordingly in the data dictionary.

REFERENCES

1. T. DeMarco, Structured Analysis and System Specification. New York: Yourdon Press, 1978
2. Computer Sciences Corporation, CSC/SD-80/6007, Mathematical Specifications for the Tracking and Data Relay Satellite System (TDRSS), S. Y. Liu, February 1980
3. --, CSC/SD-79/6040, System Specifications and Software Design for the Simulation and Error Analysis Processing of TDRSS Tracking Data, A. C. Long and M. Rajan, June 1979
4. National Aeronautics and Space Administration, Goddard Space Flight Center, X-582-76-77, Mathematical Theory of the Goddard Trajectory Determination System (GTDS), J. O. Cappellari, C. E. Velez, and A. J. Fuchs, April 1976
5. Computer Sciences Corporation, CSC/TM-80/6052, The Two-Way TDRSS Observation Model for the LSI-11/23 Microcomputer, C. Goorevich, July 1980
6. Digital Equipment Corporation, Microcomputer Processor Handbook, 1979
7. --, PDP-11 Processor Handbook, 1979
8. Computer Sciences Corporation, CSC/TM-80/6086, Orbit Determination Software Development for Microprocessor-Based Systems: Evaluation and Recommendations, C. M. Shenitz, July 1980
9. National Aeronautics and Space Administration, Goddard Space Flight Center, STDN No. 724, Tracking and Acquisition Handbook for the Satellite Tracking and Data Network, Revision 2, September 1978